PREDICTIVE DESIGN ANALYTICS FOR OPTIMAL SYSTEM DESIGN

BY

JUNGMOK MA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Associate Professor Harrison M. Kim, Chair
Professor Deborah L. Thurston
Assistant Professor James T. Allison
Assistant Professor Daniel B. Work

Abstract

"Predictive Design Analytics" proposed by this dissertation is a new paradigm to enable design engineers to extract important patterns from large-scale data characterized by four dimensions (volume, variety, velocity and veracity), and combine the extracted knowledge and its trend with complex systems optimization for various design decision making problems such as economical life cycle design, product family design and sustainable design. The goal of this research is the development of predictive design analytics methods for optimal systems design: Demand Trend Mining, Continuous Preference Trend Mining, Predictive Data-Driven Product Family Design, and Predictive Usage Mining for Life Cycle Assessment. To the best of the author's knowledge, this is one of the first attempts to provide a systematic framework of predictive analytics for design, which comprises data preprocessing, data representation, predictive analytics algorithms, mathematical formulation of design problems, and design decision making.

Demand trend mining (DTM) is developed to link pre-life (design and manufacturing) and end-of-life (remanufacturing and recycling) stages of a product for the improvement of initial product design. In order to capture hidden and upcoming trends of product demand, the algorithm combines three different models: decision tree for large-scale data, discrete choice analysis for demand modeling, and automatic time series forecasting for trend analysis. DTM dynamically reveals design attribute patterns that affect demands. A new design framework, Predictive Life Cycle Design (PLCD), is formulated, which connects DTM and optimal product design. The DTM algorithm interacts with the optimization-based model to maximize the total profit of a product through its life. For illustration, the developed model is applied to an example of smart-phone design, assuming that used phones are taken back for remanufacturing after one year. The result shows that the PLCD framework with the DTM algorithm identifies a more profitable product design over a product's life cycle when compared to traditional design approaches that focus on the pre-life stage only.

Continuous Preference Trend Mining (CPTM) is developed to generate multiple profit cycles of product design while addressing some fundamental challenges in previous studies. The CPTM algorithm captures a hidden trend of customer purchase patterns from accumulated transactional data. Unlike traditional, static data mining algorithms, the CPTM does not assume stationarity, but dynamically extracts valuable knowledge from customers over time. By generating trend embedded future data, the CPTM algorithm not only shows higher prediction accuracy in compari-

son with well-known static models, but also provides essential properties that could not be achieved with previously proposed models: utilizing historical data selectively, avoiding an over-fitting problem, identifying performance information of a constructed model, and allowing a numeric prediction. Furthermore, the formulation of the initial design problem is proposed, which can reveal an opportunity for multiple profit cycles. This mathematical formulation enables design engineers to optimize product design over multiple life cycles while reflecting customer preferences and technological obsolescence using the CPTM algorithm. For illustration, the developed framework is applied to an example of tablet PC design in the leasing market, and the result shows that the determination of optimal design is achieved over multiple life cycles.

Predictive, data-driven product family design (PDPFD) is proposed as one of the predictive design analytics methods to address the challenge of determining optimal product family architectures with large-scale customer preference data. The proposed model expands clustering based data-driven approaches to incorporate a market-driven approach. The market-driven approach provides a profit model in the near future to determine the optimal position and number of product architectures among product architecture candidates generated by the k-means clustering algorithm. Unlike discrete choice analysis models which were used in previous market-driven approaches, a market value prediction method is proposed as a dynamic model which can capture and reflect the trend of customer preferences. Prediction intervals provide market uncertainties of the dynamic profit model for product family architecture design. A universal electric motors design example is used to demonstrate the implementation of the proposed framework with large-scale data. The comparative study shows that the PDPFD algorithm can generate more profit than pure clustering based data-driven models, which shows the necessity of combining data-driven and market-driven approaches.

Predictive usage mining for life cycle assessment (PUMLCA) is developed to provide the usage modeling in life cycle assessment (LCA) which has been rarely discussed despite the magnitude of environmental impact from the usage stage. The PUMLCA algorithm can serve as an alternative of the conventional constant rate method. By modeling usage patterns as trend, seasonality, and level from a time series of usage information, predictive LCA can be conducted in a real time horizon, which can provide more accurate estimation of environmental impact. Large-scale sensor data of product operation is suggested as a source of data for the proposed method to mine usage patterns and build a usage model for LCA. The PUMLCA algorithm can provide a similar level of prediction accuracy to the constant rate method when data is constant, and the higher prediction accuracy when data has complex patterns. In order to mine important usage patterns more effectively, a new automatic segmentation algorithm is developed based on the change point analysis. The PUMLCA algorithm can also handle missing and abnormal values from large-scale sensor data, identify seasonality, formulate a predictive LCA for existing and new machines. Finally, the LCA of agricultural machinery demonstrates the proposed approach and highlights its benefits and limitations.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Harrison Kim for his excellent guidance, encouragement and support for doing this research throughout the years. This was one of my toughest journey but I could keep the momentum going based on the inspiration driven by my advisor. It is truly an honor to have Professor Kim as my academic and life advisor.

I extend my sincere appreciation to my committee members, Professor Deborah Thurston, Professor James Allison and Professor Daniel Work for their dedication and insightful suggestions that have enhanced the quality of this work. It is a great honor that my work was guided by loyal and eminent committee members.

I would like to thank Republic of Korea Army and Korea National Defense University (KNDU) for providing me an opportunity to pursue my doctoral degree. A special thank you to the sustainability research group of Deere and Company including Peter Finamore, Jeff Nelson, and Erica Knight for their valuable insights and discussions. Many thanks to the members of Enterprise Systems Optimizations Laboratory (ESOL) for their helps and advice. Thanks also go to the faculties and staffs of the Department of Industrial and Enterprise Systems Engineering for their heartfelt support.

Finally, I would like to thank my parents Murak Ma and Gabryun Kim, and my brother Jungho Ma and his family for their trust and support. Special thanks go to my wife Mihye Chun and my son Euan Ma. They always encourage me when I have tough time, motivate me when I feel lost, and show their endless support and love when I need someone. Without them, I could not have done this dissertation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Prediction, Design and Analytics: Design Engineers in the Era of Big Data

The advent of modern sensor networks and various types of web devices has brought the era of *Big Data* in the society of engineering system design. Even though the term *Big Data* is usually defined loosely in terms of quantity and diversity, it has already become essential for business models and decision making processes. Instead of simply storing data, enterprises now make great efforts to discover facts about customers and product systems from data. Naturally, design engineers are required to support data-driven decisions for the design of optimal systems not only based on traditional data but also based on new types of data. Traditional data is usually acquired from controlled environment with limited number of samples, e.g., survey and design of experiments. Traditional data is stated preference data with close customer interaction, which is designed to be tested for hypotheses. On the other hand, the new data is collected from actual behavior of customers, e.g., sales, transactions, reviews, on-line ratings, tweets, wireless sensor data, etc. The new data is revealed preference data, which is characterized as *Big Data*.

Throughout this dissertation, the term *Big Data* will not be used due to the vagueness of its definition. Instead, large-scale data characterized by 4 Vs [6] (hereinafter large-scale data) will be used in the domain of system design. The 4 Vs represent volume, variety, velocity and veracity. The volume is the amount of available data. Since data volumes keep increasing, it is difficult to define how big is really big, and it varies by industry. In terms of raw data (before preprocessing), between terabytes (10^{12} bytes) and petabytes (10^{15} bytes) of data are usually considered as big nowadays [6]. In system design, if the volume of data after preprocessing is greater than the volume of data obtained from the traditional controlled environment (hundreds or thousands at most), then it is considered to be big data by this study. The variety refers to various types of data and data sources. Data can be not only structured but also semi-structured or unstructured in the forms of on-line reviews, tweets, sensor data, transactional data, clickstreams, search queries, etc. The velocity represents the speed of data collection. In system design, it is not only the speed of data generation which is important but also capability to trace changes of underlying data patterns by collecting time series.

This implies that customers or target systems that generate data change their patterns over time such as preferences and system characteristics. The veracity denotes the level of reliability and uncertainty. For example, real-time sensor data can have missing values and abnormal values. With the dimension of velocity, forecasts are frequently required, and inevitably, forecasts have high variation (dispersion). More importantly, the veracity of data can raise the fundamental question: the context of data (e.g., clickstreams and search queries can infer the intention of purchase?). Without careful deliberation and analysis of customer behavior, extracted knowledge can be misleading.

The challenges of the four dimensions of large-scale data in the domain of system design are as follows. First, methods to handle a large data volume (dimension of volume) are needed with a proper data representation (dimension of variety). Multicollinearity, a large number of factors, computational burden and non-parametric properties are possible problems. Second, the change of underlying data patterns (dimension of velocity) should be identified and addressed in design problems. Instead of assuming steady-state processes, a trend of target information should be traced and reflected in design if it is necessary. Third, forecasting accompanies prediction intervals and uncertainties (dimension of veracity). It is important to quantify this uncertainty and support design decisions with the prediction intervals.

In order to deal with these challenges, design engineers should be equipped with tools from three critical disciplines: design (modeling and optimization), analytics (data-driven approaches or data mining), and prediction (time series modeling and forecasting). Design problems can be formulated using mathematical programming and solved with optimization techniques. Design analytics (data analytics for design) finds important hidden knowledge from large-scale data, and the identified knowledge is incorporated in design problems. The capability of prediction allows design analytics to reflect the trend of target information and to address uncertainties of future events.

1.2 Motivation

Today's highly competitive market situation and enormous data generation environment from both enterprises and customers require companies and design engineers' close adaptation to the changes of customer preferences and requirements. Business strategies and product planning are now supported by large-scale data from various sources such as social networks, sensors, clickstreams, etc. In order to accommodate the diversity and variability of customer preferences, predictive design analytics (PDA) is proposed in this dissertation.

PDA is a new paradigm to enable design engineers to extract knowledge from large-scale, multidimensional, unstructured, volatile data, and transform the knowledge and its trend into design decision making. The PDA methods encompass data-driven tools and techniques such as statistics, machine learning, data mining, and time series analysis. Since design engineers face new challenges, i.e., mining useful patterns from large-scale data and designing optimal

systems based on them, the PDA methods developed in this dissertation can shed new light on important design problems for design engineers.

1.3 Research Focus

1.3.1 Goal and Scope

This research aims to develop predictive design analytics (PDA) methods to detect useful patterns from large-scale data, and combine the extracted knowledge with system optimization for various decision making processes, e.g., economical life cycle design, product family design, sustainable design, etc. Figure 1.1 shows the structure of the overall framework.

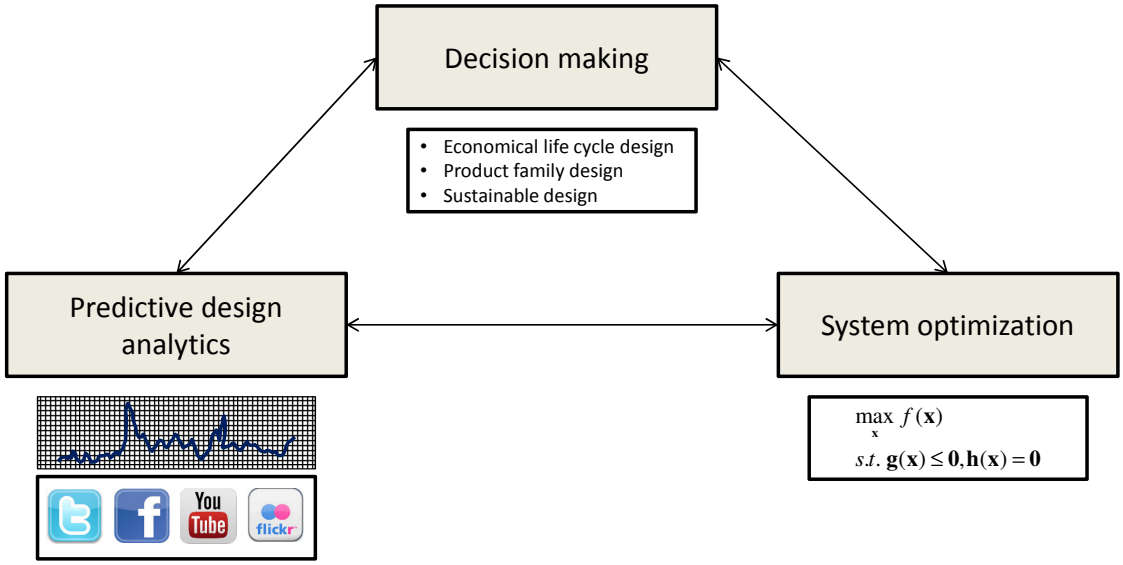


Figure 1.1: Structure of overall framework

The scope of this study comprises the developments of PDA methods and the formulation of design problems in various decision making processes while addressing how to overcome the 4 Vs of large-scale data. Figure 1.2 shows the key techniques to address some issues in large-scale data. Data-driven models or data mining based methods (e.g., supervised and unsupervised learning) are proposed to deal with the dimension of volume. A proper data representation is also discussed depending on design problems and collected data for the dimension of variety. Time series analysis is used to address the dimension of velocity (i.e., change of underlying patterns). Finally, handling missing and abnormal values (reliability) and prediction intervals (uncertainty) are discussed for the dimension of veracity. Furthermore, design problems are formulated to utilize the PDA methods and find the optimal design decisions.

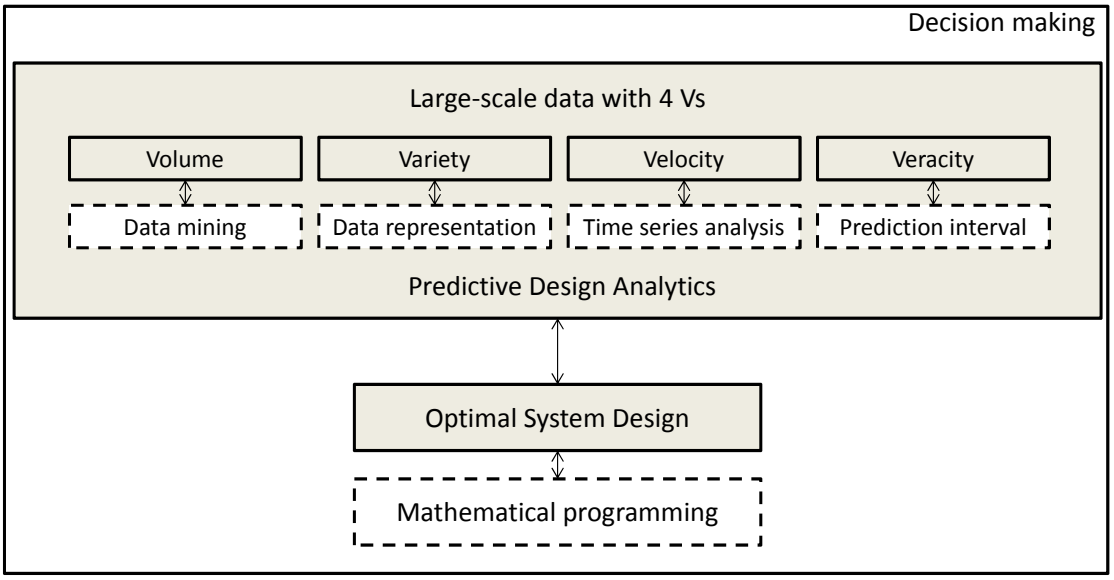


Figure 1.2: 4 Vs and proposed model

Figure 1.3 shows the flow of PDA and the focus area (dotted box). The overall procedure starts from data collection and storage from various sources such as wireless sensor network system, social networks, Web search engines, etc. The collected data should be cleaned to remove errors and the proper data representation should be determined. Depending on the collected data and design purposes, different data/trend mining techniques can be developed to detect valuable patterns or knowledge. Finally, the optimization engine finds optimal solutions based on the extracted knowledge, system models and constraints. Design decisions can be made based on the optimal solutions and sensitivity analysis. It should be noted that this study mainly focuses on the analysis of data rather than the collection and storage of data though Chapter 6 provides some basic data cleaning techniques for missing and abnormal data.

1.3.2 Research Questions

This dissertation consists of PDA methods developed for different design decisions with large-scale data. An overview of the research structure is shown in Figure 1.4. The first part (Chapters 3 through 4) of this dissertation develops PDA methods for economical life cycle design with multiple life cycles. In order to link pre-life and end-of-life decision making processes of target products, the first study (Chapter 3) investigates a method to capture hidden and upcoming trends of customer preferences and product demands from large-scale data.

Specific research questions are as follows:

- How can customer preferences information be captured for the first life and the second life of products?

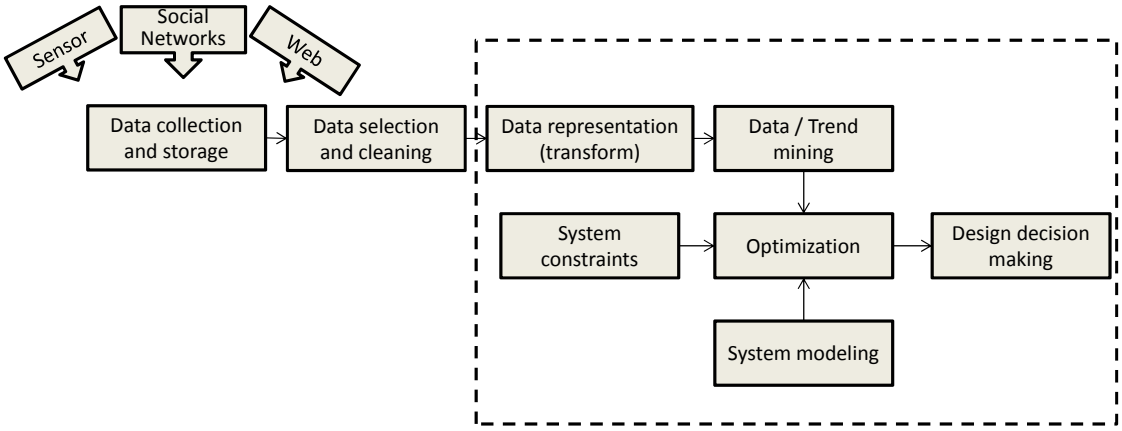


Figure 1.3: Research flow and focused area

- What are the factors which can affect the decision making process of economical life cycle design?
- How can the demand of target products be estimated from customer rating (utility) data?
- How is the pre-life and end-of-life combined profit optimization problem formulated mathematically?

The second study (Chapter 4) examines how to improve the algorithm used in the first study while identifying multiple profit cycles. This leads to the development of a new PDA method and the following research questions are explored:

- How are continuous class variables allowed in the trend mining algorithm?
- Can over-fitting problems be handled properly?
- Can the performance information of a constructed model be identified?
- How is the multiple life cycles problem formulated with the new algorithm?

The second part (Chapter 5) of this dissertation investigates data-driven and market-driven combined product family design with large-scale data. Out of the 4 Vs, the dimension of veracity is handled with prediction intervals and a large volume of data is tested for the dimension of volume. The research questions are as follows:

- What is the data representation to find the relation between product architectures and customer preferences?
- How can a future profit model be formulated and estimated with prediction intervals?
- Can multiple values for common parameters in product family design be realized?

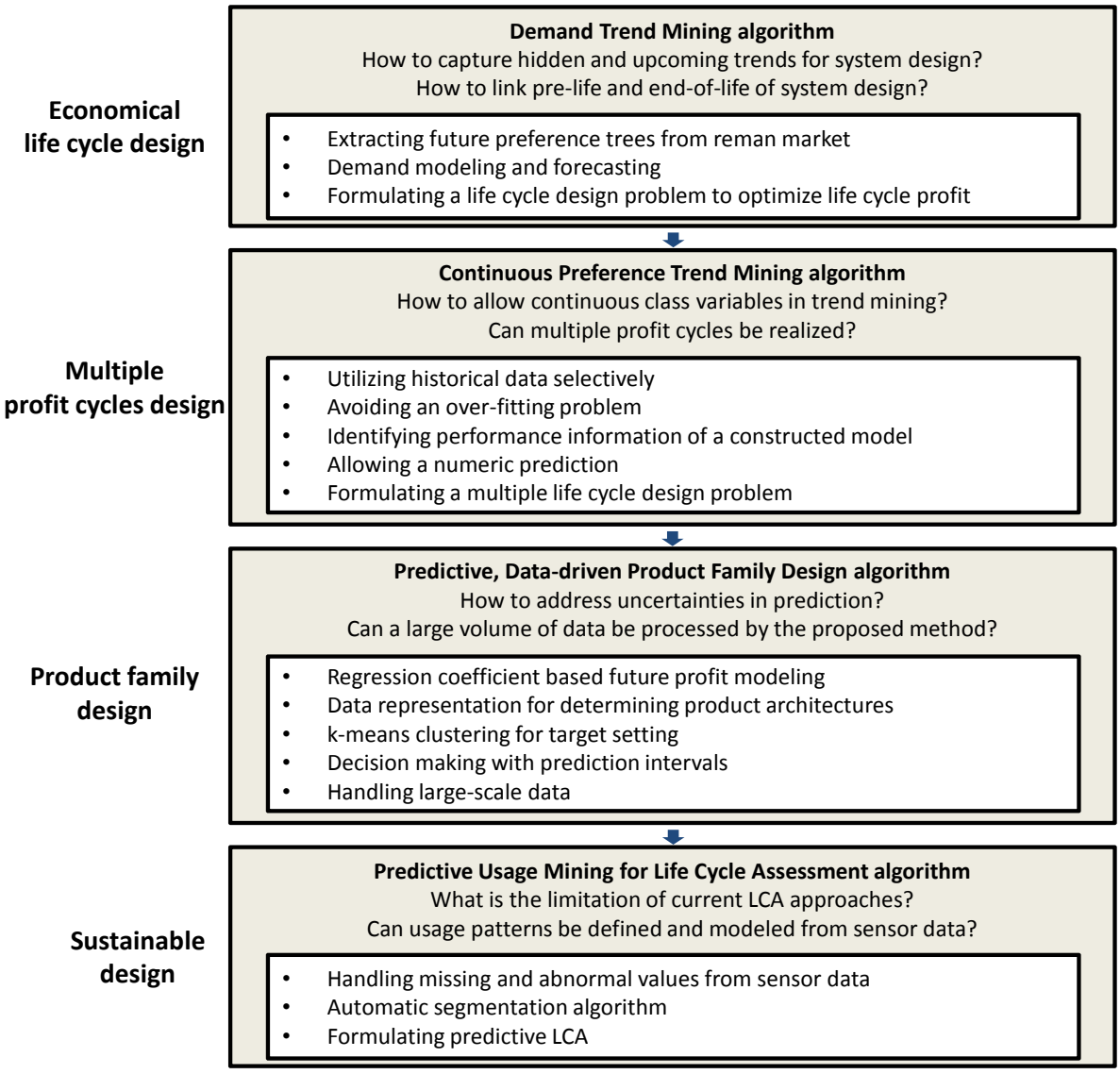


Figure 1.4: Research structure

The third part (Chapter 6) of this dissertation investigates usage pattern mining for sustainability of complex systems design with sensor data. Life cycle assessment is a core part of sustainable design and a new perspective of usage modeling is provided with large-scale sensor data. The research questions are as follows:

- Can usage patterns be modeled for life cycle assessment?
- What are the techniques for extracting patterns from sensor data?
- What is the formulation for predictive life cycle assessment in a real time horizon?

1.4 Overall Organization

The overall organization of the dissertation is illustrated in Figure 1.5. Chapter 1 introduces the motivation of predictive design analytics for design engineers who face large-scale data. The research questions are also provided for each chapter. Chapter 2 presents a review of the related literature. A survey of predictive analytics and target design areas is provided. Chapter 3 proposes a demand trend mining algorithm, which includes handling large-scale data, capturing trend and modeling demand over time. Predictive life cycle design is formulated mathematically so that pre-life and end-of-life combined profit optimization can be realized. Chapter 4 proposes a continuous preference trend mining algorithm, which can overcome some limitations of discrete trend mining algorithms. Multiple profit cycles design is formulated mathematically so that multiple profit cycles can be revealed. Chapter 5 presents a predictive, data-driven product family design algorithm, which combines data-driven and market-driven approaches. A new product family design problem is formulated mathematically so that optimal family decisions can be obtained from large-scale historical data. Chapter 6 proposes a predictive usage mining for life cycle assessment algorithm, which provides a new perspective of usage modeling in life cycle assessment. An automatic segmentation algorithm is developed for highly seasonal system data. Chapter 7 summarizes the contributions of this dissertation and discusses future work.

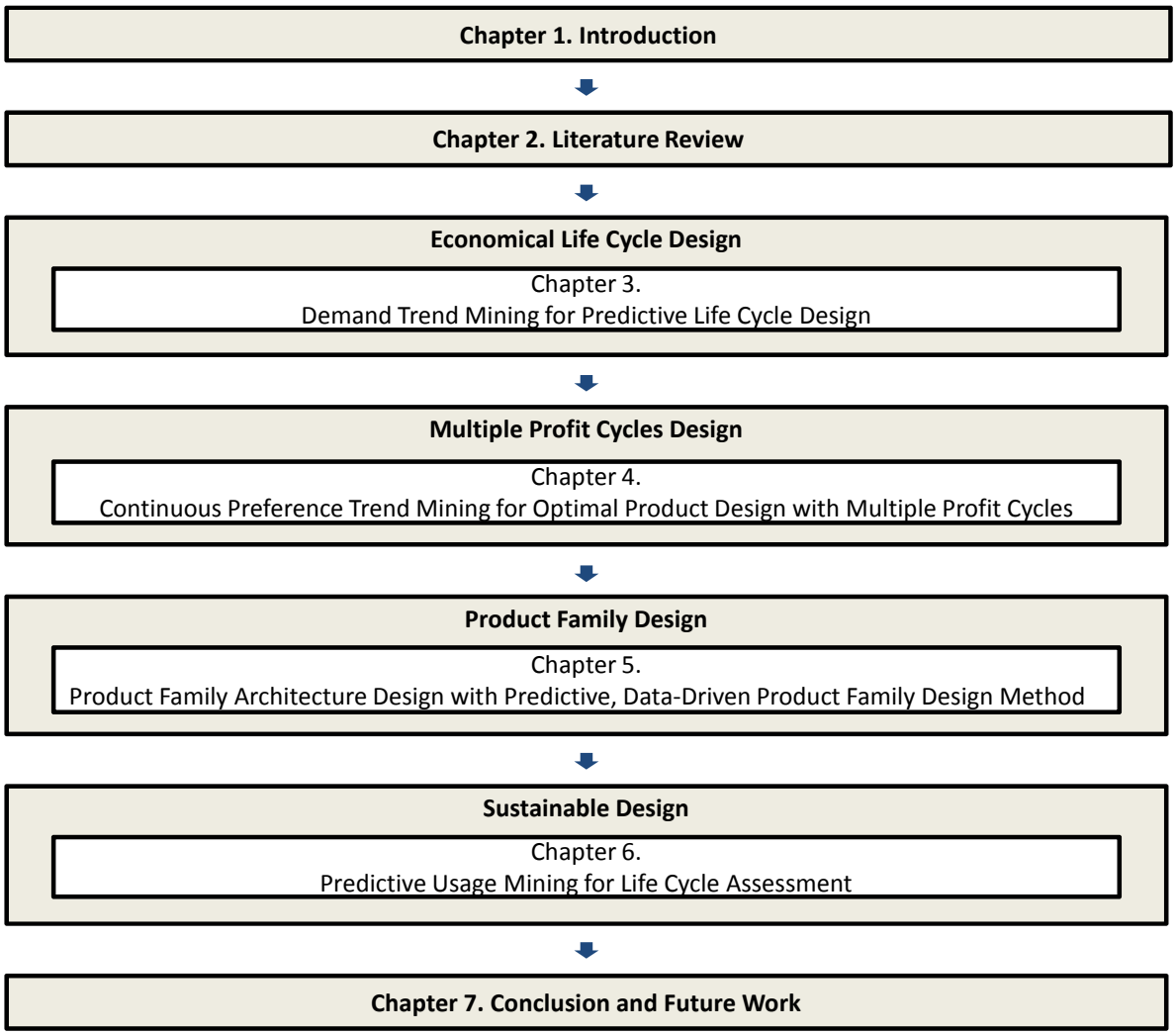


Figure 1.5: Overall organization of dissertation

Chapter 2

Literature Review

A survey of relevant studies is presented in this chapter. The review is partitioned into four sections as illustrated in Figure 2.1.

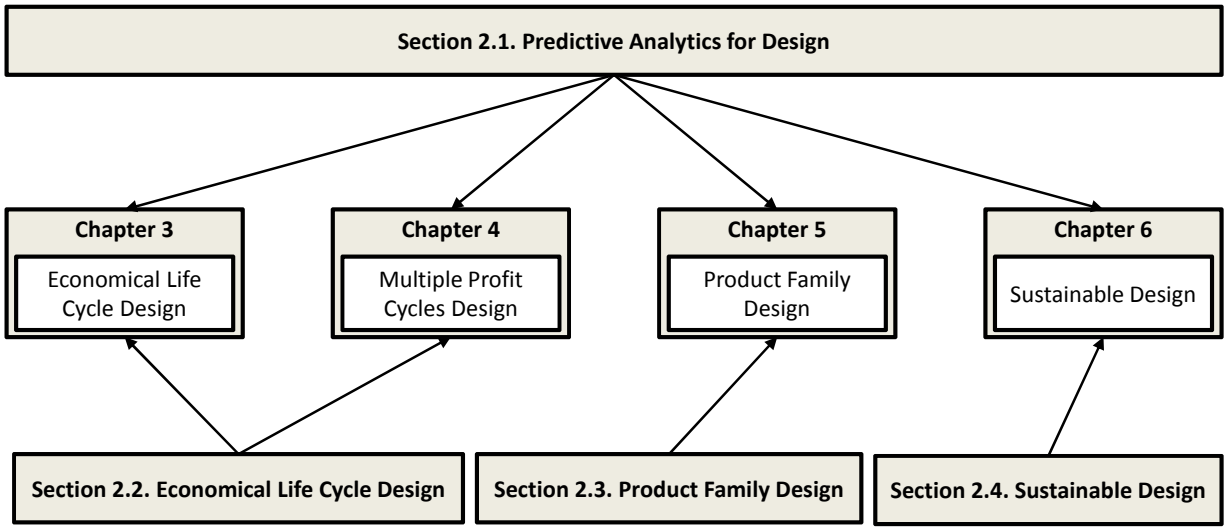


Figure 2.1: Scope of topics discussed in literature review

2.1 Predictive Analytics for Design

Predictive analytics is emerging as a new area for various businesses with data explosion. Companies such as IBM, KXEN, SAS, SAP, etc. consider predictive analytics as core tools for optimizing business processes, and universities such as Northwestern University, UC Irvine, NC state University, Depaul University, etc. have been providing degrees and certificates for predictive analytics. Google trends (http://www.google.com/trends) also show an increasing interest in predictive analytics as shown in Figure 2.2.

Due to the multidisciplinary nature and the expanding scope of predictive analytics, it is difficult to define predictive analytics but some definitions are available in the literature as follows:

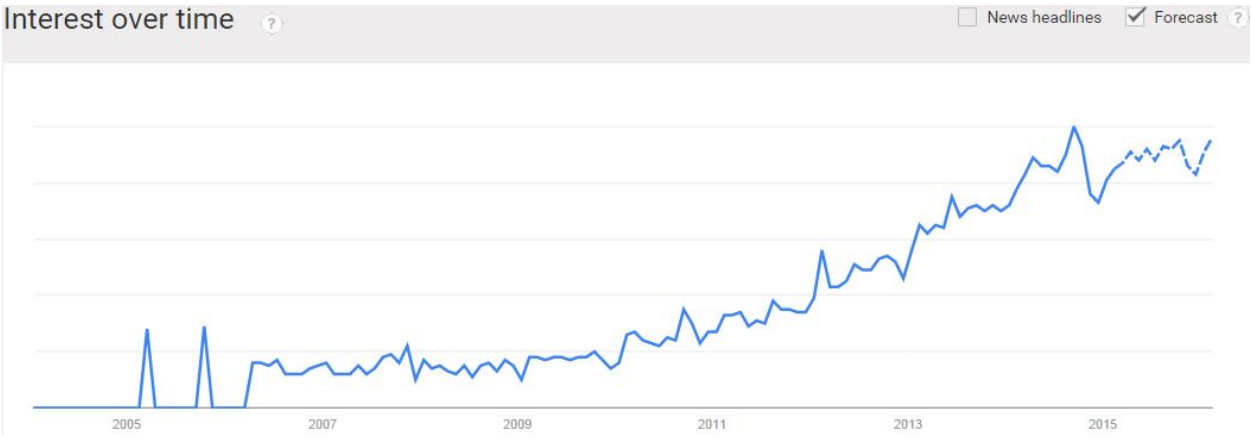


Figure 2.2: Trend of keyword “predictive analytics” (relative scale)

- “Technology that learns from experience (data) to predict the future behavior of individuals in order to drive better decisions.” [7]
- “A set of business intelligence (BI) technologies that uncovers relationships and patterns within large volumes of data that can be used to predict behavior and events.” [8]
- “Predictive analytics brings together management, information technology and modeling. It is for today’s data-intensive world. Predictive analytics is data science, a multidisciplinary skill set essential for success in business, nonprofit organizations and government.” [9]

Predictive analytics is different from query, reporting, search and visualization tools, which are considered as traditional BI technologies, in that they are inductive in nature rather than deductive [8]. That is, predictive analytics mines useful patterns from past events to predict unseen events without any presumption about data. On the other hand, the traditional BI technologies explore the data based on hypotheses and explain what the data shows. For example, predictive analytics provides a way to evaluate credit scores, detect fraudulent transactions, classify spam e-mails, provide behavioral advertising, etc., from large-scale data.

Predictive design analytics in this dissertation has two distinctive features from predictive analytics in the literature: 1) deals with both stationary and non-stationary data and 2) provides a framework to combine the methods of predictive analytics and optimal system design under various decision making processes. Most studies in the literature use the term predictive analytics and data mining (also machine learning and knowledge discovery) interchangeably. Actually, data mining is frequently introduced as main techniques for predictive analytics. However, traditional data mining techniques cannot be used as predictive models when there are underlying changes of data.

The first distinctive feature of predictive design analytics is a dynamic modeling aspect to deal with changes of data patterns. In contrast to previous studies, two types of predictions are considered. The first type is the generalization

of data. For example, without observing all the combinations of attributes (explanatory variables and independent variables), can class variables (response, output and dependent variables) from an unseen combination be predicted? Traditional data mining techniques (static machine learning models) mainly focus on this type of prediction with the implicit assumption of stationarity. That is, underlying patterns stay the same over time, which means predictive models built in the past can be used in the future. In this case, the data is called cross-sectional data. The second type of prediction is forecasting future class variables over time. For example, what is the demand of target products next month? In this case, the data is called a time series. Predictive analytics should consider both types of prediction depending on the type of data which can be cross-sectional, time-series or mixed one (i.e., time-series cross-sectional data).

Predictive trend mining (also known as change mining [10, 11] or learning concept drift [12]) is proposed to consider both types of prediction and work as a core technique for predictive design analytics. Unlike traditional static data mining models with the assumption of stationarity, the predictive trend mining models are dynamic and adaptive models that capture trend or change of target information over time.

Böttcher [10] suggested that decision trees can be built based on predicted values of interestingness measure (IM). IM is a term for "various measures devised for evaluating and ranking discovered patterns produced by the data mining process" [13]. To trace the trend of IMs, a polynomial regression model was utilized. Tucker and Kim [14] suggested the adoption of the time series analysis technique, Holt-Winters exponential smoothing model, which is a more complex modeling technique with time-variant data. Tucker and Kim showed that the trend mining technique can provide better performance than static data mining models. Support vector machine (SVM) [15] is another data mining tool that can be used in predictive trend mining. The SVM algorithm learns by example to classify different groups. Klinkenberg [12] discussed several methods to handle concept drifts based on the SVM algorithm. With concept drifts, different weighting schemes for historical data are possible, i.e., each data point over an extended period of time can be removed or utilized based on its age by allocating individual weights. Klinkenberg showed that the performance of his adaptive techniques outperformed that of simple heuristic approaches such as using all data or the most recent data in his simulated experiments.

Another important distinctive feature of predictive design analytics is the combination of predictive analytics and engineering design. To the best of the author's knowledge, this is one of the first attempts to apply predictive analytics for system design, and the combination of predictive analytics and engineering design is rarely discussed in the literature. Tucker and Kim [14] proposed the Preference Trend Mining (PTM) algorithm which can predict upcoming trends and provide a classification of design attributes as standard, non-standard and obsolete. However, the work could not be extended to general design problems because the algorithm only allowed discrete class variables and the values of design attributes were assumed to be fixed. In addition to developing methods of predictive design analytics,

this study also aims to provide a framework to combine the result of predictive design analytics and optimal system design.

In summary, predictive design analytics consists of *data analytics* which analyzes and extracts important patterns from large-scale data, *design analytics* which focuses on the data related to the domain of engineering system design, and *predictive analytics* which serves as a predictive model for both cross-sectional data and time-series data. In the following sections, some design areas are explored, which can be improved by predictive design analytics methods.

2.2 Economical Life Cycle Design

Design for life cycle or life cycle design is a design paradigm that enables design engineers to close the loop of a product life cycle and to manage its life cycles. It focuses on the fact that decisions made at the design stage affect all phases of a product's life cycle (i.e., material extraction, manufacturing, usage, and end-of-life recovery and disposal). Since it is hard to predict the product's end-of-use state (e.g., life cycle length, product condition, product recovery decision, preference of remanufactured product, etc.), predictive design analytics can be particularly beneficial for the area of product life cycle design and recovery by analyzing large-scale data from the users, original equipment manufacturers (OEMs), markets, and the public over a product's life. It is critical to link product's pre-life (design and manufacturing) and end-of-life for closing the loop of a product life cycle.

Recovery of end-of-life products (especially electronics) has become an urgent problem that requires design engineers' attention due to multiple reasons. The first reason is the fast growing e-waste stream. The U.S. Environmental Protection Agency (EPA) estimated that 2.37 million short tons of electronic products were ready for the end-of-life processes in 2009 [16]. That is almost 50% higher than in 1999, and only 25% of them were gathered for recycling. The second reason is the fact that electronic products contain toxic and hazardous materials [17]. Lead, mercury, nickel, and palladium are examples that present negative environmental and human health effects. Reckless landfills are not an optimal solution. A third reason for recovering these products is that electronic products also contain reusable and valuable resources, such as gold, copper, tin, nickel, etc. [17]. Efficient and systematic methods to recover the reusable parts and resources are needed. Fourth, more regulations and responsibilities have been enforced. The countries in the European Union have already begun adopting product take-back policy (Extended Producer Responsibility, EPR) since 1991 [18]. The U.S. has also introduced more EPR laws recently compared to 2006 [19]. Fifth, "green consumers" [20] give more pressure to companies regarding their "green" image. Now, consumers' increased awareness of sustainability is a critical factor in determining the demand of target products. Lastly, product recovery and recycling are known to reduce fuel consumption and landfill space, and provide substantial benefits to the environment [21].

Some OEMs such as Caterpillar, Xerox, and Sony have shown that a proper recovery system of their end-of-life products not only extends their products' lives and gives some environmental benefits, but also allows for multiple profit cycles [22, 23, 24]. These OEMs consider the end-of-life stage as the "re-life" stage and return take-back components to "same-as-new" condition to customers. The re-life processes or recovery options include reuse, repair, refurbishment, cannibalization and recycling. By introducing remanufactured products back to the market, companies can find new profit opportunities and establish environmentally friendly image.

The evidence from these OEMs indicates that the construction of a system for recovery can be a hidden source of profit. However, many factors should be considered in order to determine the profitability of the recovery system. Possible sources of uncertainties are the product's life, the state (product condition) after its life, available quantities for recovery, the reliability of a remanufactured product, customer preferences, and technological obsolescence.

Moreover, many studies showed that the initial design of a product would determine $70 \sim 85\%$ of total life cycle cost and environmental impact [25, 26, 27], so the selection of initial design attributes is very important. Life cycle design is aimed at proactively dealing with economic and environmental issues during the early design stage when the potential for affecting results is the greatest. Since little research has been conducted for the economic perspective in comparison with the matured environmental perspective over entire life cycle [28, 29], the economic side of product design is investigated in this dissertation, which is called *economical* life cycle design.

Some researchers [30, 31, 32] have developed a holistic design approach that considers various concerns of all life cycle stages in an integrated manner. However, a more popular approach has been to develop design principles for improving a specific life cycle stage. Design for recovery, design for remanufacturing, design for disassembly, and design for recycling are among the principles of life cycle design. In design for end-of-life, researchers seek to identify optimal product design to reduce the cost of recovery and/or increase the profit associated with recovery.

Rose et al. [33, 34] suggested a classification scheme for helping designers predict appropriate recovery strategies for a product, so that the designers can redesign products to move toward a higher level of reuse. Mangun and Thurston [35] developed a model for designing a product portfolio that incorporated part reuse through refurbishment. Given multiple market segments with varying requirements for environmental impact, production cost, and reliability, they attempted to determine the optimal product design for each segment in order to maximize the total utility of the portfolio. Kwak and Kim [36, 37] introduced a framework for analyzing how product design affects end-of-life recovery and what architectural characteristics are desirable for higher recovery profit.

One limitation of these previous methods, however, is that the design implications on the pre-life and end-of-life stages have been considered separately. Product design has been optimized for each of the stages, but not for the stages together due to the lack of available demand forecasting models. Two exceptions can be found in Zhao and Thurston [38] and Kwak and Kim [39]. Both developed a mathematical model to determine the optimal product design

that maximizes the profits from both initial sales and end-of-life recovery. They showed that the total profit can be maximized when both ends of the product life cycle are considered at the same time. However, the prediction and reflection of demand trends in the market were not incorporated. Predictive design analytics will capture customer preferences from historical data and help to maximize the total profit from the entire life cycle of a product.

2.3 Product Family Design

Product family design represents designing "a set of products that share one or more common elements (e.g., components, modules, and subsystems)" in order to satisfy various market applications [40]. The product family design paradigm was successfully implemented by companies such as Sony, Hewlett Packard, Black & Decker, Volkswagen, and Rolls Royce [41, 40]. One of the important tasks in this complex engineering design problem is the determination of optimal product family architectures [42]. The product architecture is "the arrangement of functional elements to the physical building blocks" [43] and works as a target (e.g., performance requirements) of engineering design for product variants [42]. The products of interest in this dissertation are products or parts that can be highly shared by many other products, including universal motors in power tools and home appliances, engines in on and off-road vehicles, batteries in electronics, etc. These products should satisfy a wide variety of different customers' requirements.

Recent advances in product family design were discussed in [44] from customer needs, functional requirements, design parameters, process variables to logistics variables. Basically, there are two approaches in product family design to utilize a product platform ("the set of parameters and/or features that remain constant" [3]): module-based and scale-based product family design [45]. Module-based product family design represents building related products using functional modules from the platform while scale-based product family design represents designing related products by varying (e.g., stretch or shrink) scaling variables while making common parameters constant. Examples of both approaches can be found in [45]. This study focuses on multiple-platform scale-based product family design. The multiple platforms represent multiple values for common parameters. Multiple-platform design was studied using a heuristic approach with clustering analysis based on sensitivity analysis [46] and an information theoretical approach [47]. It is shown in this dissertation that a simple clustering algorithm can be used to explore the possibility of multiple platforms with given common parameters. Some previous works [48, 49, 47] discussed product family design with unknown common parameters.

In multiple-platform scale-based product family design, product family architecture design is a target setting problem for product variants [42]. It is also a positioning problem of a product family into different market segments or clusters of customer preferences [44]. Clustering techniques can be used to find the number of product variants which encompass the maximum possible customer preferences [44].

Previous studies mainly focused on two directions. The first direction is the development of commonality indices which are used to assess success of family design: degree of commonality index [50], total constant commonality index [51], product line commonality index [52], comprehensive metric for commonality [53], etc. The second direction is the development of solution techniques for family design problems: generalized reduced gradient [3], sequential linear programming [48], nonlinear programming [54], genetic algorithm [55], etc. as single-stage, two-stage, and multi-stage approaches [45, 56, 57].

In order to utilize large-scale data with market uncertainty, two emerging approaches are reviewed. First, data-driven approaches represent modeling techniques to capture important information and its trend from data. For example, customers' various requirements for a product can be widespread in the space of product design. The objective is to determine the number and position of product architectures in order to satisfy customers' requirements. However, only data-driven approaches might generate geometrically meaningful results. For example, one architecture can be the optimal solution based on the selected information criterion (model fitting function with penalization of complex models) but it might end up with an inferior solution from the perspective of markets. With the guidance of market-driven approaches, data-driven approaches can produce a meaningful result for decision makers. Once architectures are determined then clusters based on the architectures can be interpreted as market segments. For example, finite types and models of cars are being manufactured to cover their market segments or different ranges of customer preferences.

Second, market-driven approaches represent profit modeling techniques, which evaluates product architecture candidate sets in terms of profit. When the number of product architectures is increased, the fixed costs, and price that customers are willing to pay will be increased accordingly. Only market-driven approaches in product family design relies on information of market segments. If pre-defined market segments are not available or segments can be changed over time due to the volatility of customer preferences, determining product architectures and their specifications can be a challenging decision making process. Data-driven approaches can be a solution of this issue by extracting necessary information (i.e., clusters of customer preferences) from data.

Market-Driven Product Family Design

A market-driven approach in product family design aims to integrate market considerations with product family architecture design [44]. In order to translate customer requirements into design requirements (including functional requirements), quality function deployment and its variant techniques were used [41, 44]. Discrete choice analysis [58, 59, 60] is a popular model in engineering design problems to map design attributes into market share estimation.

de Weck et al. [42] proposed a methodology that determines the optimum number of product platforms to maximize product family profitability with simplifying assumptions. The methodology is divided into family level (plat-

form architecting) and variant level (product optimization) design. First, market segments and corresponding market leaders should be identified. The number of market segments is set to be the maximum number of product platforms. Second, the design variable set, objective function, and demand equation for a single market segment needs to be established. Since each market segment is assumed to have a unique performance requirement, each segment represents each platform. Third, product architectures should be optimized for a given performance requirement, and the profit of the product family can be estimated. de Weck et al. [42] assumed that all the necessary information of the first and second step is given so that the determination of number of platforms is the only decision variable in the family level (i.e., no architecture specification).

Kumar et al. [61] developed market-driven product family design, which expands the demand modeling part of de Weck et al. [42]. First, the methodology starts from the creation of market segments. All the necessary information such as required performance, price, customer demographics, and competitors are identified. After that, a nested logit demand model is built. The role of the demand model is to determine the market share of each market segment with specified product performance, customer demographics, and price. Second, models for product performance and costs need to be built. These models make trade-offs between cost and performance in the demand model. Third, optimal product specifications and a number of platforms are identified to maximize the overall profits. Similar to the work of de Weck et al. [42], product specifications for each segment were given (as different constraints).

These market-driven approaches extend the scope of product family design by introducing a profit model as an objective function. The number of product family architectures was considered as one of the design variables to maximize the profit function. However, information about market segments was assumed to be given instead of derived. Moreover, the profit model based on discrete choice analysis is static, which means a built model in the past can be used anytime in the future. This study aims to relax the stationarity of profit modeling.

Data-Driven Product Family Design

Agard and Kusiak [62] introduced the possible usage of the data mining based methodologies for product family design. Given that customer demographics and functional requirements are available, clustering techniques can be applied to group similar customers so that a representative customer can be identified. Also, functional requirements can be associated with each other in order to find dependencies using association rule mining techniques. Moon et al. [63] proposed that data mining techniques can identify a platform with variants and unique modules. Association rule mining captured associated rules from product functions, and these rules were clustered as modules using fuzzy c-means clustering. Tucker et al. [64] developed a product family optimization model with ReliefF attribute weighting and X-means clustering techniques. The X-means clustering gave the number and specifications of architectures, and the ReliefF provided the importance of each design attribute in the optimization model. Chan et al. [65] proposed

fuzzy clustering to group customer requirements as market segments. The center points of market segments were used for the development of product variants.

These studies showed that market segmentation can be realized automatically by clustering techniques instead of being assumed to be given or resorting to experts' opinions. However, they did not consider market so it is possible to have sub-optimal solutions in terms of profit. Moreover, previous studies involved small data sets (e.g., 50 in [65] and 1000 in [64]).

Predictive design analytics provides a data-driven and market driven combined method for product family design, which can mine important market information from accumulated large-scale data sets without pre-defined market segments, and capture a trend of customer preferences over time with the consideration of its uncertainty. Since product family architecture design is a complex and difficult task, predictive design analytics can shed new light on this problem for design engineers.

2.4 Sustainable Design (Life Cycle Assessment)

In Section 2.2, the basic concepts of design for recovery, design for life cycle, and design for remanufacturing are introduced with an economic perspective. The original goal of these design methods is to accomplish green, environmentally friendly, sustainable design. For sustainable design, environmental life cycle assessment is an essential tool to evaluate success of sustainability.

Life cycle assessment (LCA) is an analytical assessment tool to quantify environmental impact of a product or system [66, 67]. The potential environmental impact can be generated from all the stages of a product, i.e., manufacturing, usage, maintenance, and end-of-life. The LCA approach provides a holistic and systematic way to manage data associated with the target product. With the popularity of sustainable design and environmentally conscious design, LCA studies have been reported for various materials, electronics, automobiles, and complex systems [29].

The LCA framework [68, 69] consists of goal and scope definition, inventory analysis (LCI, life cycle inventory), impact assessment (LCIA, life cycle impact assessment), and interpretation. The goal and scope definition is the phase that defines the purpose, target systems or products, the level of sophistication. The LCI is the phase that defines the system boundaries and the flow diagrams with unit processes (e.g., extraction of oil, refining, production of electricity, etc.). The main result from the LCI is the inventory table which quantifies inputs (e.g., raw material, land, energy, etc.) from and outputs (e.g., pollutants such as CO_2, SO_2, NO_x, etc.) to the environment. The LCIA is the phase that translates the inventory table into relevant impact categories (e.g., carcinogens, climate change, acidification, etc.) and quantifies the environmental impact using weighting and normalization. The interpretation is the phase that evaluates the results from the LCIA and makes recommendations of the LCA study.

A couple of well-established impact assessment methods were proposed by organizations and researchers, e.g., Eco-Indicator series, IPCC (Intergovernmental Panel on Climate Change) Global Warming Potential (GWP) series, ReCiPe, Ecosystem Damage Potential, etc. Two popular methods are introduced as follows.

Eco-Indicator 99

The Eco-Indicator 99 [2] is designed to provide a single score (points) which can be interpreted as the environmental impact of a product. Figure 2.3 shows the procedure to calculate the indicator. First, based on the inventory analysis, inputs (resources and land use) and outputs (emissions) are identified. Second, the items of the constructed inventory table are classified as proper impact categories and mapped to the three damage categories such as resources, ecosystems and human health. Third, the final indicator is calculated as a single score using the weighted sum of the three damage categories.

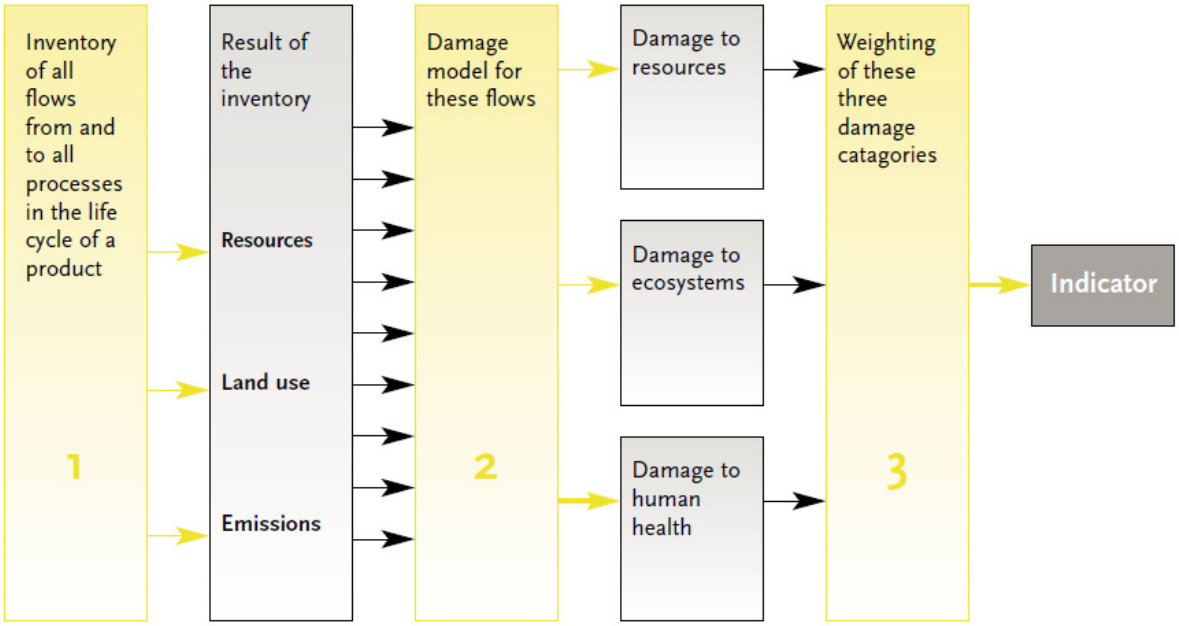


Figure 2.3: Eco-Indicator 99 framework from [2]

Global Warming Potential (IPCC 2007)

The IPCC 2007 method (https://www.ipcc.ch) provides a quantified measure of greenhouse gases' (e.g., carbon dioxide, methane, nitrous oxide, etc.) global warming potential (GWP). Unlike the Eco-Indicator 99, weighting is not included in this method. Instead, carbon dioxide is used as the reference gas, which has a GWP of one. The IPCC 2007 method provides the conversions of other greenhouse gases based on the reference gas and GWP values are ap-

plied to units of mass (e.g., kilograms CO_2 equivalents). A GWP is calculated over time horizons of 20, 100, and 500 years. For example, the 100 year GWP of nitrous oxide is 298, which represents that the unit amount of nitrous oxide emitted to air can cause 298 times more greenhouse effect than the unit amount of carbon dioxide over 100 years.

A set of software programs is available to implement LCA studies such as SimaPro (http://www.pre-sustainability.com), Gabi (http://www.gabi-software.com), openLCA (http://www.openlca.org), Team (http://ecobilan.pwc.fr/en/boite-a-outils/team.jhtml), GREET (http://greet.es.anl.gov/), etc. They provide an interface to connect environmental databases and apply the impact assessment methods.

Although the LCA approach is mature and has become a widely used method in various industries, it is usually *static* in that time is not considered in the assessment with the implicit assumption of steady-state processes. The necessity of considering time in LCA was discussed in literature. Reap et al. [70] provided insightful reviews on the temporal aspects of LCA. Temporal factors such as different rates of emissions over time and seasonal variation of their impacts can influence the accuracy of LCA. Levasseur et al. [71] showed that the inconsistency in time frames can affect LCA results significantly. Memary et al. [72] demonstrated that changes of environmental impact over time are useful information for assessing future technology and options. Collet et al. [73] presented a method to find the most critical flows of information based on dynamic inventory data (i.e., LCI level) and sensitivity analysis. In addition to the aspect of time, spatial variation is another contributor that can significantly affect the accuracy of LCA [70]. Local, regional and continental differences can cause the different result of LCA.

Among the life cycle stages of a product, the manufacturing stage, which is the chosen stage in the majority of LCA studies, can be considered as a one-time event, i.e., time-independent event. Although the dynamic inventory approach [73] attempted to relax this (e.g., the impact from material x or process y can be changed over time), the inventory data is considered constant in this study. On the other hand, the usage stage (with maintenance and end-of-life stages) is a time-dependent event, which means the lifespan of a product has a large impact on LCA. Many studies showed that the majority of environmental impact can come from the usage stage over life cycle (e.g., more than 60% for cars [74], more than 80% for off-load machinery (product of interest in this study) [75], and 80$\sim$90% for some small electronics [76]).

Even though the importance of the usage modeling has been recognized among LCA researchers and practitioners, it is rarely discussed in literature. LCA studies in literature usually utilized a constant rate [77, 78, 75, 1, 79] of usage information (hereinafter constant rate method) with the implicit assumption of steady-state processes (e.g., average fuel consumption rate in kg/hr, fixed operating hours per month, etc.). This method is simple and easy to apply, but if data has complex patterns (e.g., trend, seasonality and segments), the prediction accuracy of the constant rate method can be significantly reduced. The constant rate method only allows us to calculate life cycle impact in a nominal time horizon, e.g., 10 years as a whole instead from October 2014 to December 2024. This can be an important issue to

policy makers and manufacturers when they want to estimate the environmental impact of the future.

One exception is Telenko and Seepersad [76] who proposed a usage context modeling technique in LCA using Bayesian network models. The usage context includes human, situational, and product variables. Based on a predefined probabilistic network of relevant usage patterns (e.g., weather→ usage of electric kettle with probability of x), a usage profile and its variability can be modeled as a form of distribution. However, in order to apply this approach, causal relationships among different usage contexts should be known, which is expressed as a probabilistic network. For example, the usage of agricultural machinery (e.g., crop sprayer, harvester, nutrient applicator, etc.) can be affected by a various usage context (e.g., weather, soil, experience of farmers, price of fuel and crops, machine deterioration). It will be difficult to correlate these variables with specific usage information (e.g., diesel fuel consumption and operating hours). Furthermore, Telenko and Seepersad [76] did not consider time in LCA.

Predictive design analytics can provide a new usage modeling technique from sensor data. Based on large-scale time-stamped data, target usage information can be modeled and predicted for the LCA of complex systems. Predictive LCA in a real time horizon will estimate the environmental impact of target systems more accurately than the conventional usage modeling method.

2.5 Discussion

This chapter provides the background of predictive analytics. In the literature, predictive analytics has been getting more attention and being perceived as a new business intelligence model. Predictive design analytics in this dissertation emphasizes that proposed predictive trend mining techniques can deal with not only cross-sectional data but also time-series cross-sectional data while traditional data mining techniques are static and limited to cross-sectional data. Moreover, it has been rarely discussed on how to apply predictive analytics for design problems. This study also provides a framework to utilize the result of predictive design analytics in the formulation of design problems. Some design areas are reviewed and analyzed in order to determine whether predictive design analytics can provide new opportunities to improve the available design methods. Following chapters will discuss the development of predictive design analytics methods for those design areas with the four dimensions of large-scale data.

Chapter 3

Demand Trend Mining for Predictive Life Cycle Design

In this chapter[1], a new demand modeling technique, Demand Trend Mining (DTM), is proposed for Predictive Life Cycle Design. The first goal of this chapter is the development of the DTM algorithm for predicting future demands. In order to capture hidden and upcoming trends of product demand, the algorithm combines three different models: decision tree for large-scale data, discrete choice analysis for demand modeling, and automatic time series forecasting for trend analysis. DTM dynamically reveals design attribute patterns that affect demands. The second goal is the new design framework, Predictive Life Cycle Design (PLCD), which connects DTM and data-driven product design. This new optimization-based model enables a company to optimize its product design by considering the pre-life (manufacturing) and end-of-life (remanufacturing) stages of a product simultaneously. The DTM algorithm interacts with the optimization-based model to maximize the total profit of a product. For illustration, the developed model is applied to an example of smart-phone design, assuming that used phones are taken back for remanufacturing after one year. The result shows that the PLCD framework with the DTM algorithm identifies a more profitable product design over a product life cycle when compared to traditional design approaches that focuses on the pre-life stage only.

3.1 Introduction

Product design analytics or data-driven product design is emerging as a promising area by bridging benefits of large-scale data and product design decisions. With the popularity of social network and web devices, a large volume of data which has a characteristic of complexity, timeliness, heterogeneity, and lack of structure [82] are being generated every day. Although the necessity of large-scale data analytics for product design is now being recognized broadly, only a few researchers have attempted to analyze large-scale data in the context of product and design analytics [83, 14, 84]. This study proposes Demand Trend Mining (DTM) as one of the analysis tools for large-scale data in order to capture the trend of demand as a function of design attributes. DTM is a dynamic and adaptive model in that it mines the underlying changes of concept drift from time series data and builds a predictive model based on the changes. The model shows that it can realize predictive life cycle design which encompasses both the *pre-life* (i.e., manufacturing)

[1] Presented in [80] and published in [81].

and *end-of-life* (i.e., remanufacturing which is used as an umbrella term for reuse, reconditioning, refurbishment, and cannibalization. and recycling) stages.

Remanufacturing has been a new profit opportunity for original equipment manufacturers (OEMs). Caterpillar, Xerox, and Sony are among the OEMs who have successfully taken this new opportunity [22, 23, 24]. In remanufacturing, used products are restored to a like-new condition and are given another life in the market. Remanufacturing can bring larger profits over the lifespan of a product from an initial investment at low additional costs, typically 40% to 65% less than new product costs because it reutilizes the materials and the value added to a product in its initial manufacturing [85, 86].

Remanufacturing also enables OEMs to improve their environmental performance. As awareness of environmental issues increases, pressure from the public and policymakers have prompted OEMs to be responsible for the environmental impacts of their products. OEMs now need to extend their environmental efforts to encompass the entire life cycle of a product, from cradle (raw material extraction) to grave (end-of-life disposal). By remanufacturing a product, OEMs can reduce waste and minimize the need for raw material to make new products. It is known that remanufactured products (hereinafter *reman product*) can save up to 90% of the environmental impact of entirely new products [87, 24].

In order for successful remanufacturing, design for life cycle (or life cycle design) is key for OEM remanufacturers. Product design determines not only the current profit from the pre-life stage but also the future profit from the end-of-life stage [88, 36, 38]. Therefore, to maximize the total profit from the entire life cycle of a product, OEM remanufacturers must optimize their design decisions considering both stages together.

The main challenge in life cycle design is that there is a significant time gap (i.e., usage-life) between the pre-life and end-of-life stages. As illustrated in Figure 3.1, suppose that the decision maker is at time $t^{prelife}$ (design stage), and the selling point of new product is t^{first}. In this research, it is assumed that the time gaps between $t^{prelife}$ and t^{first}, and t^{eol} and t^{second} are known. Also, it is assumed that the usage-life is h, remanufacturing will occur at time t^{eol}, and the remanufactured products will be sold at the market at time t^{second}. For instance, the typical usage-life of cell phones and laptops is known as 1.5 years [89] and 4 years [90], respectively. Considering rapid changes in technology and customer preferences, such a time gap between pre-life and end-of-life stages implies that life cycle design should consider and satisfy two sets of customer needs at the same time, i.e., needs for new products at the present and needs for reman products in the future. Although many demand models have been presented for capturing current demands at the new-product market (hereinafter *new market*), very few models are available for forecasting future demands at the remanufactured-product market (hereinafter *reman market*). Moreover, little research has been presented that combines a dynamic demand model with life cycle design, which considers the time gap and transforms a trend of customer preferences to projected demands.

Another challenge is uncertainty of returned products in terms of quantity, timing, and condition. Figure 3.1 shows material flow starting from material extraction to part manufacturing, product assembly, recovery and disposal. The scope of the problem is clearly defined using solid arrows. In this study, recovery options are categorized as material, part, and product levels. Product level recovery (e.g., reuse and reconditioning) only requires some minor value-added operations including polishing, cleaning, and lubricating. Part level recovery (e.g., cannibalization and refurbishment) needs disassembly as well as parts conditioning and change. Material level recovery (e.g., recycling) is usually conducted by recyclers and raw materials are recovered by shredding and refining. There are three possible cases that require corporations' end-of-life decisions: initial returns, returns within warranty period, and take-back program. The initial returns are caused by changes of purchase decisions in a short period of time. The returns within warranty period are induced by defects in any time. The focused case, take-back program, aims at boosting sales with re-purchasing contracts of sold products within specified period. In this case, the amount and condition of returned products should be considered in a model.

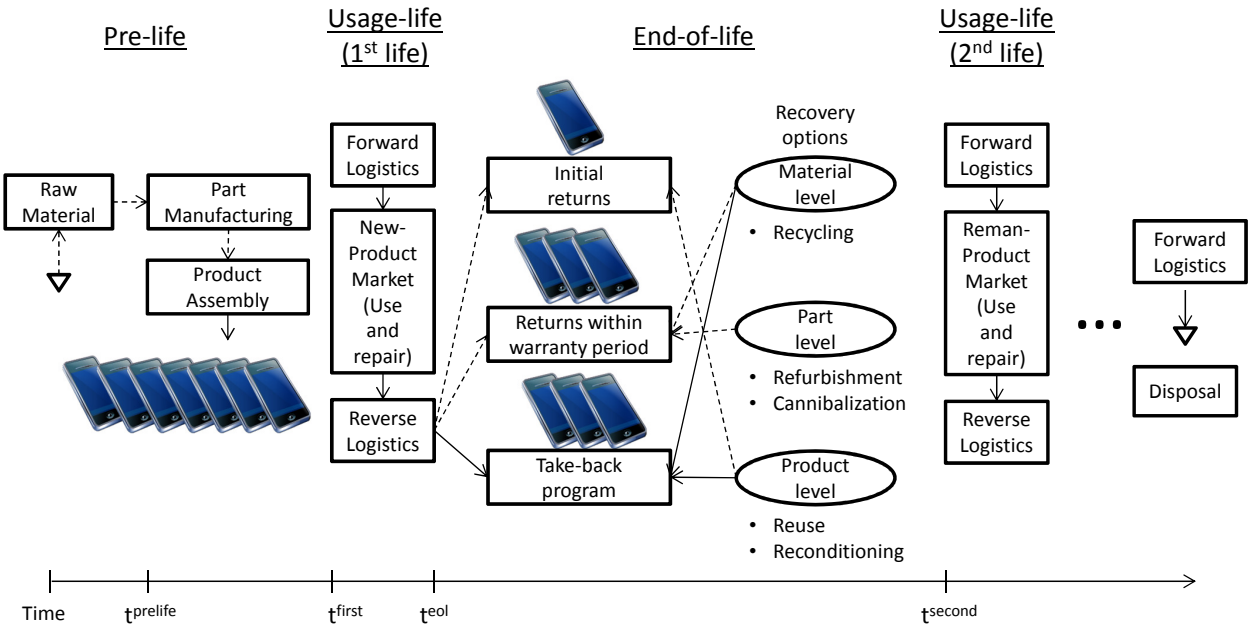


Figure 3.1: Closing the loop of product life cycle and scope of the problem (solid arrow)

Modeling demand and customer preferences are critical for assessing the profit of a product. Under the framework of Predictive Life Cycle Design (PLCD), the decision maker should consider two markets at the initial design stage, i.e., the current market for new products and the future market for reman products. Considering the time gap between pre-life and end-of-life stages, customer preferences in the two markets are likely to be different. DTM thus aims to construct two demand models: one for new products and the other for reman products.

Two widely used demand analysis techniques in product design are discrete choice analysis (DCA) [59, 60] and conjoint analysis [91, 92]. While both techniques can capture customers choice behavior and model related demands, they resort to direct customer interactions (e.g., survey) and have a limited capability to use large-scale data due to the statistical assumptions [93].

Decision tree in data mining is an alternative model for customer preferences in product design. Since the decision tree algorithm can deal with large-scale massive data, it was proposed to generate product concepts for design engineers [93, 94]. However, very little research was conducted on demand analysis with the decision tree in the field of product design and other system design [14, 95].

In order to capture trends of demand, dynamic demand models should be constructed instead of static demand models. Dynamic models do not assume that demand models that were once built would remain the same over time. Böttcher [10] suggested that decision tree can be built based on predicted values of interestingness measure (IM). IM is a term for "various measures devised for evaluating and ranking discovered patterns produced by the data mining process" [13]. To trace the trend of IMs, a polynomial regression model was utilized. Tucker and Kim [14] suggested the adoption of the time series analysis technique, Holt-Winters exponential smoothing model, which is a more complex modeling technique with time-variant data. However, there exist different classes of exponential smoothing, which means the Holt-Winters model is just one of its family and design engineers should choose right one among them. At the same time, designers are required to determine many different parameters and initial states for the Holt-Winters model. The DTM algorithm adopted the Hyndman's automatic time series forecasting algorithm [96, 97]. This algorithm includes the automatic optimization process for model selection, parameter setting, and initial state estimation with the *innovations* formulation of state space models.

The proposed DTM combines the merits of aforementioned three different models: DCA for demand modeling, decision tree for large-scale data, and automatic time series forecasting for trend analysis. The decision tree algorithm, C4.5, models customer preferences from large-scale data, and by formulating a class variable as utility, the resulting decision tree models can estimate market shares from the DCA, specifically logit choice probability in the multinomial logit (MNL) model. Automatic time series forecasting provides predicted IMs, and trend reflected demand is estimated from the target time decision tree.

Table 3.1 provides a summary of MNL and C4.5 [14, 98]. The MNL model starts from a random utility model where the true utility consists of the observable utility and the unobservable random part. In the MNL model, the random part is assumed as independent and identically distributed extreme value, and the choice probability is given by the logit choice probability. The C4.5 algorithm is based on the information theory. Entropy, a measure of disorder or complexity, is calculated, and the decision tree is built in the direction of minimizing the entropy.

The DTM algorithm which is depicted in Figure 3.2 tackle some challenges systematically: extracting valuable

Table 3.1: Overview of MNL and C4.5

	MNL	C4.5
Assumption	- Random Utility Model $U_{nj} = V_{nj} + \epsilon_{nj}$ $\epsilon_{nj} \sim iid\ extreme\ value$ *j* : choice alternative *n* : decision maker	- Information Theory (Information Entropy) $Entropy(D) = -\sum_{i=1}^{k} P_i \cdot \log_2 P_i$ *D : data set* *k : number of class variables within the data set* P_i : probability of class variable i
Choice Probability & Split Criterion	- Logit Choice Probability $P_{ni} = \frac{\exp(V_{ni})}{\sum_j exp(V_{nj})}$ Decision maker n choose alternative i over alternative j (i ≠ j)	- Gain Ratio $Gain\ Ratio(X) = \frac{\text{Entropy(D)} - \sum_{j=1}^{n} \frac{\lvert D_j \rvert}{\lvert D \rvert} \cdot Entropy(D_j)}{-\sum_{j=1}^{n} \frac{\lvert D_j \rvert}{\lvert D \rvert} \cdot \log_2 \frac{\lvert D_j \rvert}{\lvert D \rvert}}$ *X : attribute* *n : number of outcomes for a given attribute*

knowledge from large-scale data, building a demand model from the mined knowledge, and predicting a target demand in the future. The requirements for overcoming the challenges include 1) utilization of large-scale data, 2) estimation of demands, and 3) realization of demand trends over time. In order to fulfill these requirements, the DTM algorithm utilizes and combines three different models: discrete choice analysis (DCA), decision tree, and automatic time series forecasting. If $t = 1$ to $t = n$ data are available and $t = h$ ahead demand is needed, then DTM provides a way to estimate demand at $t = n + h$ as shown in Figure 3.2. To combine the DCA and decision tree, a class variable of a decision tree model is proposed to be expressed as utility. Also the concept of generational difference is adopted for a prevention of missing values and smooth forecasting in product design.

Using the DTM algorithm, Predictive Life Cycle Design (PLCD) can be finally implemented. The proposed PLCD framework enables design engineers to optimize target product design by considering the pre-life and end-of-life stages of a product simultaneously. The identified product design will maximize the total profit over the entire product life cycle. Figure 3.3 provides an overview of the PLCD framework. The dotted box represents the DTM model. The remaining components represent the optimal life cycle design or optimization-based model. The framework optimizes the product attributes as well as the selling prices and production quantities of new and reman products. For illustration, the developed model is applied to an example of smart-phone design, assuming that the available products from initial sales of the pre-life will return for remanufacturing after one year of usage, according to a take-back contract.

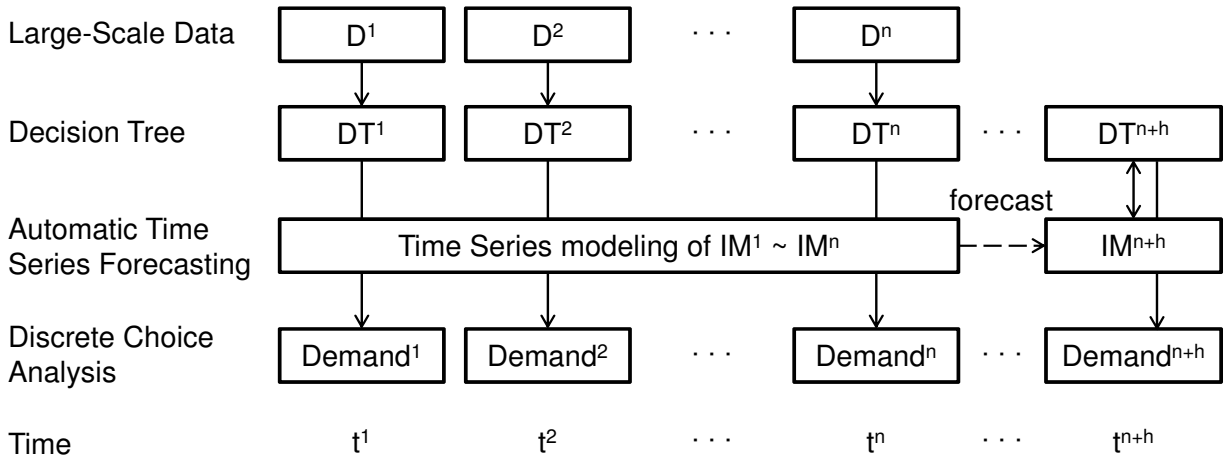


Figure 3.2: Demand trend mining algorithm

The rest of this chapter is organized as follows. Section 3.2 describes the detailed steps of overall methodology. Section 3.3 presents an illustrative case study of smart-phone design. Section 3.4 concludes the chapter with suggestions for future research.

3.2 Methodology

This section describes detailed steps for DTM and PLCD. Figure 3.4 shows the overall framework of PLCD which has two components: DTM and optimal life cycle design. Although the general description of the DTM algorithm is described in Figure 3.2, Figure 3.4 provides more detailed steps of DTM, especially in the framework of PLCD.

3.2.1 Modeling of Demand Trend

As illustrated in Figure 3.1, suppose that the decision maker is at time $t^{prelife}$, and the selling point of new product is t^{first}. It is assumed that the time gaps between $t^{prelife}$ and t^{first}, and t^{eol} and t^{second} are known. Also, it is assumed that the usage-life is h, remanufacturing will occur at time t^{eol}, and the remanufactured products will be sold at the market at time t^{second}. Thus, the PLCD framework starts from DTM which estimates the market demands at time t^{first} and time t^{second} for new and reman products, respectively. The DTM algorithm in Figure 3.2 is divided as 3 Steps in the following subsections in detailed description. Step 2 covers decision trees and automatic time series prediction, which are components of Preference Trend Mining. The demand modeling with discrete choice analysis is depicted in Step 3.

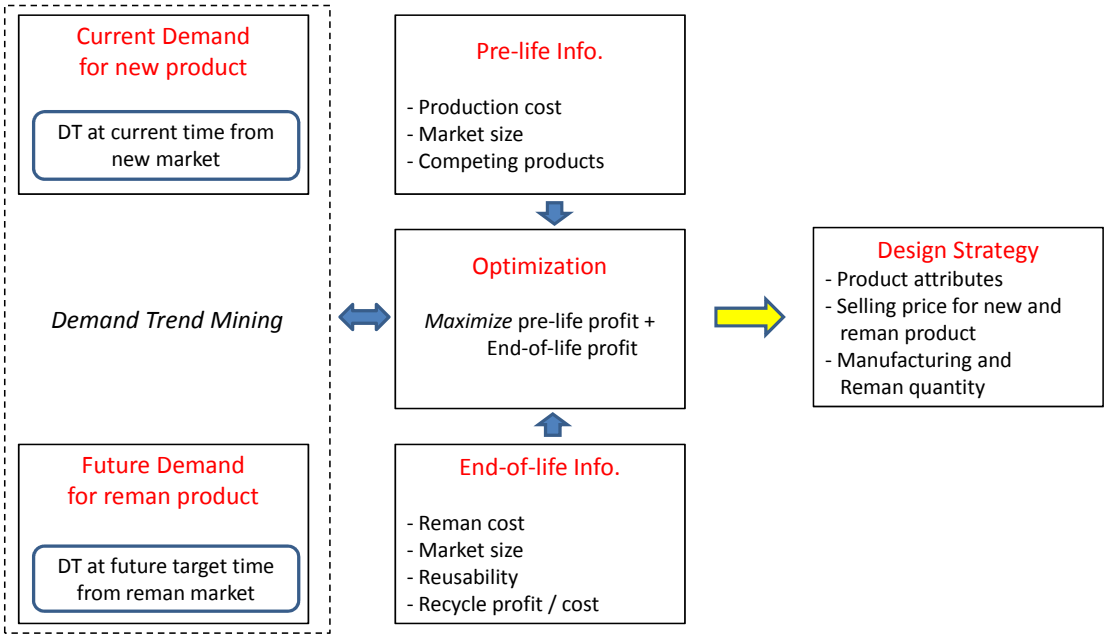


Figure 3.3: Summary of PLCD framework

Step 1: Data Collection

In the first step, two data sets are collected to capture trends of demand in the market. First, the customer preference data for new products are collected at the current time $t^{prelife}$ in Figure 3.1. This data is used for capturing market demand at the pre-life design stage. Second, the historical and the current preference data for reman products are collected to predict market demand at the end-of-life stage. The preference data from time t^1 to $t^{prelife}$ in the reman market are used to mine underlying demand trends and estimate the market demand at time t^{second}.

Table 3.2 shows the basic data structure with an example of smart-phone design. The data comprises of two parts: a set of attributes and a class variable. Attributes are product features, and class variables are outputs or responses that we are interested in. In this research, the degree of customer preference or the customer utility on a discrete scale is used as the class variable. It can be either stated data from a survey or revealed data from on-line reviews. By having utility as the class variable, demand modeling is allowed in Step 3. Each attribute has its own levels; for example, the attribute camera pixel has two different levels, e.g., 8 or 16-MP.

In the case of attributes with significant improvement in their values, it is represented in a relative scale using the concept of generational difference [99]. The generational difference can be acquired by comparing the generational gap between the target part and the latest cutting-edge part which corresponds to the maximum generation. For example, if 16-MP is the latest generation, then the generational difference is 0. If 8-MP is the previous generation, the generational difference is 1. The advantages of the generational difference include the prevention of missing values

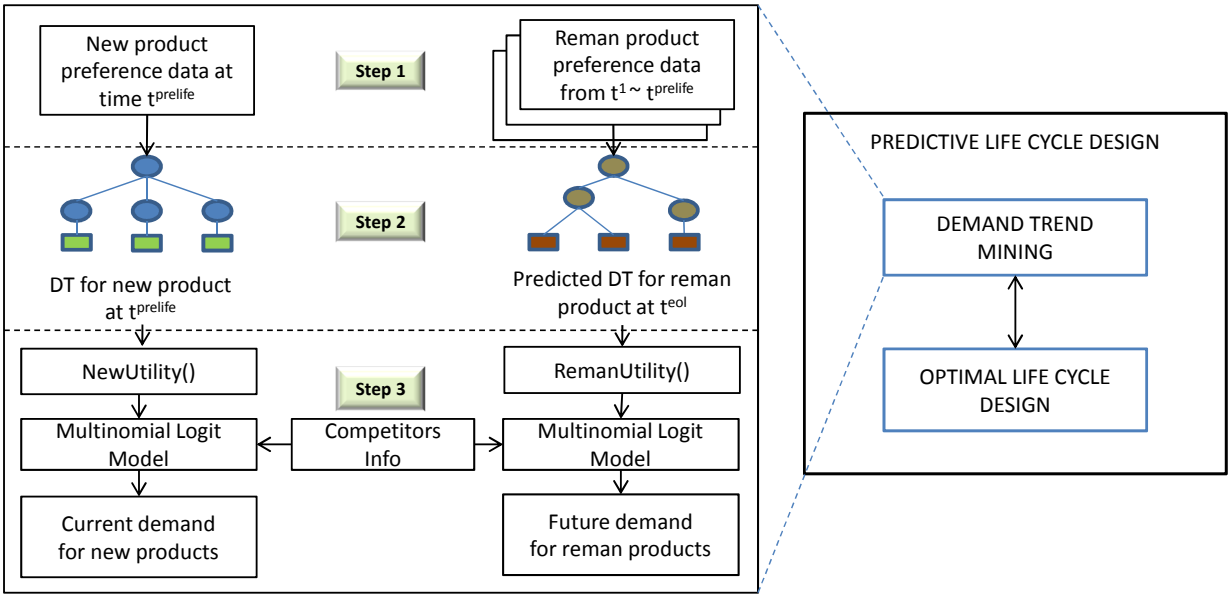


Figure 3.4: Framework of PLCD

over time and the allowance of forecasting without specific levels so that emerging trends can be captured with various levels. The original Preference Trend Mining proposed by Tucker and Kim [14], which will be discussed in the next step, was not intended to deal with various levels, as the algorithm used fixed levels over time.

Table 3.2: Data structure (with example of smart-phone design)

Smart-Phone Attribute										Class
New product Price		Reman product Price		Screen size		Memory		Camera Pixel		Utility
\$199	Y_{11}	\$99.5	Y_{21}	2.8″	X_{11}	2 (16GB)	X_{21}	1 (8MP)	X_{31}	1
\$299	Y_{12}	\$149.5	Y_{22}	3.5″	X_{12}	1 (32GB)	X_{22}	0 (16MP)	X_{32}	2
\$399	Y_{13}	\$199.5	Y_{23}	5.3″	X_{13}	0 (64GB)	X_{23}			3
										4

Step 2: Preference Trend Mining

In the second step, the data sets collected in Step 1 are analyzed in order to reveal the link between product attributes and customer utility. For new and reman products, different analyses are conducted. The data for new products is analyzed using Quinlan's C4.5 decision tree algorithm [100] which is a static model. The generated decision tree can be transformed into a set of decision-tree-based rules, i.e., NewUtility(). Each path of the decision tree expresses a decision rule and given an attribute combination, the decision-tree-based rule provides an estimate of utility.

The time series data for reman products is analyzed using the revised Preference Trend Mining (PTM) algorithm adopted by Tucker and Kim [14]. The algorithm generates a predicted decision tree for the future time t^{eol}, which can provide a set of decision rules, i.e., RemanUtility(). Algorithm 1 shows the pseudo code for the revised PTM. S_T is the time series data (from time t^1 to $t^{prelife}$) for reman products and X is the set of attributes. The revised PTM algorithm is similar to the C4.5 algorithm in that it builds the decision tree based on the interestingness measure (IM). In both algorithms, the attribute with the maximum IM becomes the node for branching. The difference is in how to calculate the IMs.

Unlike the C4.5 algorithm using one aggregated data set, the revised PTM algorithm forecasts the IMs of the future time from the historical time series data. In Algorithm 1, PTM starts from finding the IMs of all attributes X from all previous data (line 3). Then, there are processes to predict the IMs at t^{eol} using the IMs from t^1 to $t^{prelife}$ and assign the attribute with the maximum IM as the root node of the tree (line 5). The levels of the attribute then become branches. For each branch, the same processes are repeated for remaining attributes; i.e., PTM checks which attribute has the maximum IM at t^{eol} and iteratively splits a decision tree until it reaches termination criteria. After identifying all the leaf nodes, the algorithm returns the predicted decision tree.

Algorithm 1 Preference Trend Mining revised from [14]

```
1:  procedure PTM(S_T)
2:      while Termination criteria are met do
3:          Find IM(X) for S_T and Forecast IM(X) at t^eol
4:          If IM(X_i) = MAX IM(X) at t^eol
5:          Then X_i = root node, X_i levels = branches
6:          Find IM(X) for S_T and Forecast IM(X) at t^eol given selected branches
7:          If IM(X_i') = MAX IM(X) at t^eol
8:          Then X_i' = child node, X_i' levels = branches
9:          Repeat 6, 7, 8
10:     end while
11:     Result class variable = leaf node
12:     return Predicted decision tree
13: end procedure
```

To apply the revised PTM algorithm, three issues should be clarified. First, the decision maker should decide the IM to use. The IMs that are well known and widely used include Shannon's entropy, gini index, information gain, and

gain ratio. Depending on the data and its characteristics, each measure has its own pros and cons [101]. In this study, the gain ratio was selected following the C4.5 algorithm although the approach can be generalized with the other IMs.

The second issue is about the forecasting engine for the IM prediction. Hyndman's exponential smoothing [96] and the Box-Jenkins model [102, 103] are among the most popular and widely-used methods for time series forecasting. In the Hyndman's exponential smoothing, the time series data can be decomposed into four components, i.e., trend, seasonal, cycle, and irregular error. A total of 30 mathematical models are available, and the best model can be obtained using automatic time series forecasting algorithm [96, 97]. The Box-Jenkins model is another popular option. It applies an autoregressive moving average (ARMA) or an autoregressive integrated moving average (ARIMA) to fit the time series data. Exponential smoothing has value in that it is relatively simple and easy to understand though there is no general consensus about which one has a better prediction accuracy [104, 105]. In this research, Hyndman's exponential smoothing model, specifically the automatic time series forecasting method, is chosen as the forecasting engine.

The difference between the Holt-Winters model in the original PTM algorithm [14] and the automatic time series forecasting [96, 106] in the DTM algorithm is that the former is just one of exponential smoothing family and requires many of user inputs, but the latter allows automated model selection, and parameters and initial state estimation among 30 different linear and nonlinear models for design engineers.

Termination criteria in decision-tree generation is another important issue. If all class variables are distributed homogeneously and no valid split is found, the process can be stopped. If the leaf node is reached and the class variables are not distributed homogeneously, the path can be removed or the dominant class variable over time can be selected.

Step 3: Demand Modeling

The decision trees obtained in Step 2 provide two sets of decision rules, NewUtility() and RemanUtility(). The decision rules give estimates on customer utility that corresponds to a set of design attributes. NewUtility() gives the utility estimates in the current new market, and RemanUtility() gives the estimates in the future reman market.

Once customer utilities for a specific product and its competing products are given, it is possible to estimate the market share of each of the products. In this research, logit choice probability of the multinomial logit (MNL) model [58] is used as shown in Equations (3.1) and (3.2), where l and m are the product choices available in the new and reman markets, respectively; Y_{ij} is a vector of binary variables representing price related (Y_{1j} for price of a new product, Y_{2j} for price of a reman product) product attributes and their levels; X_{ij} is a vector of binary variables representing component related product attributes and their levels; MS^{new} and MS^{reman} are the sizes of new and reman markets, respectively; $D^{new}(Y_{1j}, X_{ij})$ and $D^{reman}(Y_{2j}, X_{ij})$ are market demands for new and reman products, respectively .

$$D^{new}(Y_{1j},X_{ij}) = \frac{\exp(NewUtiliy(Y_{1j},X_{ij})}{\sum_1^l \exp(NewUtiliy_l(Y_{1j},X_{ij}))} MS^{new} \tag{3.1}$$

$$D^{reman}(Y_{2j},X_{ij}) = \frac{\exp(RemanUtiliy(Y_{2j},X_{ij})}{\sum_1^m \exp(RemanUtiliy_l(Y_{2j},X_{ij}))} MS^{reman} \tag{3.2}$$

3.2.2 Optimal Life Cycle Design

The optimal life cycle design model is the optimization engine for PLCD. Table 3.3 shows the problem statement of the optimization model. With the aim to maximize the pre-life and end-of-life profits together, the model identifies the optimal product design as well as optimal production strategies at the pre-life and the end-of-life stages (i.e., the quantities and selling prices of new and reman products). The model assumes that the new products sold at time t^{first} are all taken back for recovery after h years at time t^{eol}. A certain percentage of the initial selling price, $\varepsilon \cdot P^{new}$, is paid for the take-back. It is also assumed that the returned end-of-life products are all recovered by either remanufacturing or recycling. Customer abuse and product reliability can affect the availability of remanufacturable products. Based on the product condition, only working products are allowed for remanufacturing. During the remanufacturing operation, no loss in yield or no scrap is assumed. Also, upgrades of parts are not considered. In other words, products are remanufactured maintaining their initial design from the pre-life stage.

Objective Function

The objective function of the model is given in Equation (3.3). It aims to maximize the total life cycle profit, i.e., sum of profits from the pre-life and end-of-life stages. Equation (3.4) formulates the total profit from the pre-life stage, i.e., the profit from making and selling Q^{new} units of new products at the current time $t^{prelife}$. Equation (3.5) formulates the total profit from the end-of-life stage. It mainly consists of three parts: cost of taking back $Q^{takeback}$ units of end-of-life products, profit from remanufacturing Q^{reman} units of end-of-life products, and profit from recycling $Q^{recycle}$ units of products. Since the end-of-life profit occurs at the future time t^{eol}, an annual interest rate α is applied to discount the value.

$$Maximize \quad f^{prelife} + f^{eol} \tag{3.3}$$

$$f^{prelife} = (P^{new} - C^{new})Q^{new} \tag{3.4}$$

Table 3.3: Optimal life cycle design model

Objective Function	*Maximize* (pre-life profit + end-of-life profit)
Decision Variables	- Product attributes
	- Quantity of products to be manufactured and remanufactured
Constraints	- Design attributes uniqueness
	- No excess fulfillment of products
	- Take-back program
Given Info	- NewUtility() and RemanUtility()
	- Manufacturing, remanufacturing and recycling cost, and recycle profit
	- Market size for new and reman products, and competing products
	- Reusability or reliability of components
Assumptions	- Accumulated preference data are available
	- Remanufacturing and recycling are possible recovery options
	- No loss in yield in the recovery operation
Type of Problem	Mixed Integer Non-Linear Problem

$$f^{eol} = \frac{1}{(1+\alpha)^h}[(-C^{takeback} \cdot Q^{takeback} + (P^{reman} \cdot Q^{reman} - C^{reman} \cdot Q^{reman})) + (P^{recycle} - C^{recycle})Q^{recycle}] \quad (3.5)$$

Equations (3.6) and (3.7) represent the prices of new and reman products as a function of the price related decision variable Y_{ij}. Equations (3.8) through (3.11) formulate the unit processing costs of manufacturing and recovery activities. In Equations (3.8) and (3.10), both manufacturing and remanufacturing costs are affected by binary decision variables, X_{ij}. If product attribute i ($i \in I$) has the level of j ($j \in J$), Y_{ij} equals 1; otherwise, it equals 0. ε in Equation (3.9) denotes the take-back cost parameter and $C^{privacyprotection}$ represents the cost related to activities of privacy protection (e.g., data cleaning or scrubbing).

$$P^{new} = \sum_j P_{1j}^{new} \cdot Y_{1j} \tag{3.6}$$

$$P^{reman} = \sum_j P_{2j}^{reman} \cdot Y_{2j} \tag{3.7}$$

$$C^{new} = \sum_i \sum_j C_{ij}^{manufacturing} \cdot X_{ij} + C^{forwardlogistics} \tag{3.8}$$

$$C^{takeback} = \varepsilon \cdot P^{new} + C^{reverselogistics} + C^{sorting} + C^{privacyprotection} \tag{3.9}$$

$$C^{reman} = \sum_i \sum_j C_{ij}^{reconditioning} \cdot X_{ij} + C^{forwardlogistics} \tag{3.10}$$

$$C^{recycle} = C^{recycling} + C^{forwardlogistics} \tag{3.11}$$

Constraints

Equations (3.12) through (3.20) show the constraints of the model. Equation (3.12) imposes that each product attribute i has an attribute level j. Equation (3.13) constrains the production quantity of new products, Q^{new}, in such a way that they are always less than or equal to the demand for them, $D^{new}(Y_{1j}, X_{ij})$. As described in the previous section, the demand is obtained by the decision-tree-based rules from DTM. Similarly, Equation (3.14) constrains the production quantity of reman products, Q^{reman}.

$$\sum_j Y_{ij} = 1, \sum_j X_{ij} = 1, Y_{ij}, X_{ij} \in \{0,1\} \tag{3.12}$$

$$Q^{new} \leq D^{new}(Y_{1j}, X_{ij}) \tag{3.13}$$

$$Q^{reman} \leq D^{reman}(Y_{2j}, X_{ij}) \tag{3.14}$$

Equation (3.15) formulates that available products from the new products sales at the first-life stage will be taken back for recovery at the end-of-life stage. ρ denotes the take-back loss parameter due to the customer abuse. Equation (3.16) constrains that all the returned products are recovered either by remanufacturing or recycling.

$$Q^{takeback} = \rho \cdot Q^{new} \tag{3.15}$$

$$Q^{takeback} = Q^{reman} + Q^{recycle} \tag{3.16}$$

Equation (3.17) refrains Q^{reman} from exceeding the available amount of remanufacturable products, $A(t^{eol})$. Equa-

tion (3.18) estimates $A(t^{eol})$, where it is determined by the multiplication of $Q^{takeback}$ and remanufacturability, $\delta(t^{eol})$, i.e., the probability that a product is still reusable and remanufacturable at the end-of-life stage. In Equation (3.19), $\delta(t^{eol})$ is defined as the multiplication of each part's reliability, $\gamma_j(t^{eol})$, at time t^{eol}. Because a part's reliability differs by design decisions, $\delta(t^{eol})$ is formulated as a function of X_{ij}. Finally, Equation (3.20) shows the variable conditions for production quantities.

$$Q^{reman} \leq A(t^{eol}) \tag{3.17}$$

$$A(t^{eol}) = Q^{new} \cdot \delta(t^{eol}) \tag{3.18}$$

$$\delta(t^{eol}) = \prod_i (\sum_j \gamma_j(t^{eol}) \cdot X_{ij}) \tag{3.19}$$

$$Q^{new}, Q^{reman} \in \text{nonnegative integer} \tag{3.20}$$

3.3 Illustrative Example: Smart-Phone Design

3.3.1 Overview

As the waste stream of discarded mobile phones grows rapidly, recovery of used phones has become an important issue in recent years. Mobile phones are known to have a relatively short life cycle, approximately 1.5 years [89]. In 2009, the U.S. Environmental Protection Agency (EPA) estimated that Americans discard approximately 129 million mobile devices every year, of which only 8% are recycled properly [16]. This implies not only an environmental problem but also a missing profit opportunity. According to the EPA, "recycling one million cell phones can save enough energy to power more than 185 U.S. households with electricity for a year." ReCellular, Inc is another testimony of profitable recovery. According to the Wall Street Journal [85], "ReCellular resold 5.2 million mobile phones in 2010, up from 2.1 million five years earlier, and its revenue was $66 million."

This section illustrates the PLCD framework with an example of smart-phone design. Suppose that there is an OEM smart-phone manufacturer that operates a one-year take-back program; they make and sell new products, and after one year, they take back the products for remanufacturing. For such take-back, it is assumed that the company returns 15% of the new-product price to the customer. To maximize the total profit from manufacturing and remanufacturing, the company aims to optimize their product design considering changing trends in the market. This section shows that the PLCD framework with DTM can serve their needs effectively and demonstrates that the company can achieve greater profit by adopting the model. To be specific, there are five product attributes that the company wants to optimize: selling prices of new and reman products, screen size, memory size, and camera pixels. Depending on

which attributes are chosen, the product would have different production costs and reliability, and different profits at the pre-life and end-of-life stages. Table 3.4 and 3.5 present assumptions on production costs and part reliability for attribute choices.

Table 3.4: Assumptions about manufacturing and remanufacturing cost

	Manufacturing								Remanufacturing							
	Screen			Memory			Camera		Screen			Memory			Camera	
	X_{11} (2.8″)	X_{12} (3.5″)	X_{13} (5.3″)	X_{21} (16GB)	X_{22} (32GB)	X_{23} (64GB)	X_{31} (8MP)	X_{32} (16MP)	X_{11} (2.8″)	X_{12} (3.5″)	X_{13} (5.3″)	X_{21} (16GB)	X_{22} (32GB)	X_{23} (64GB)	X_{31} (8MP)	X_{32} (16MP)
Cost($)	26	36	48	30	38	52	18	38	3.5	3.7	4	2.3	2.5	2.9	3	3.2

Table 3.5: Assumptions about part reliability after one year

Screen			Memory			Camera Pixel		
2.8″	X_{11}	0.95	16GB	X_{21}	0.9	8MP	X_{31}	0.92
3.5″	X_{12}	0.92	32GB	X_{22}	0.9	16MP	X_{32}	0.88
5.3″	X_{13}	0.88	64GB	X_{23}	0.9			

3.3.2 Demand Trend Mining

To apply the DTM algorithm, two sets of customer preference data are required: one for the current new market and the other for the future reman market. The former is collected at a single time point $t^{prelife}$ and used for estimating market demand at time t^{first}. The latter, on the other hand, is collected over multiple time points from t^1 though $t^{prelife}$ and used for capturing future demand at t^{eol}. In this study, preference data were artificially generated. A total of 216 samples were simulated for each time point. The data for reman market was simulated as ten time-stamped data with six-month intervals; in other words, preference data reflecting market trends over the last five years were collected over ten time points, t^1 to t^{10}. Here, t^{10} represents the current time $t^{prelife}$, i.e., t^{10}= $t^{prelife}$. Since the time gaps between $t^{prelife}$ and t^{first}, and t^{eol} and t^{second} were very short for the simplicity, the historical data was used for the prediction of demands at t^{12} with a one-year take-back program.

The data structure was the same as shown in Table 3.2. Each sample represented a specific combination of design attributes and the corresponding class variable (i.e., customer utility). As discussed in Section 3.2, all variables were

defined as discrete variables. Table 3.2 shows design candidates of each variable.

In order to obtain decision rules, $NewUtiliy(Y_{1j}, X_{ij})$ at $t^{prelife}$ (= t^{10}) and $RemanUtiliy(Y_{2j}, X_{ij})$ at $t^{prelife+2}$ (= t^{12}), the C4.5 and PTM algorithms were applied to the new and reman market data, respectively. Weka 3.6.5 [107] and R 2.14.0 [108] were used for the decision tree induction and automatic time series forecasting. The resulting rules are given in Figure 3.5 and 3.6. Each path in Figure 3.5 and 3.6 represents a decision rule for a utility estimation. For example, in Figure 3.5, one can estimate that if the selling price of a new product is \$199, the camera resolution is 8-MP, and the memory is 16-GB, the screen size is 2.8-inch then the corresponding customer's utility is 2 out of 4.

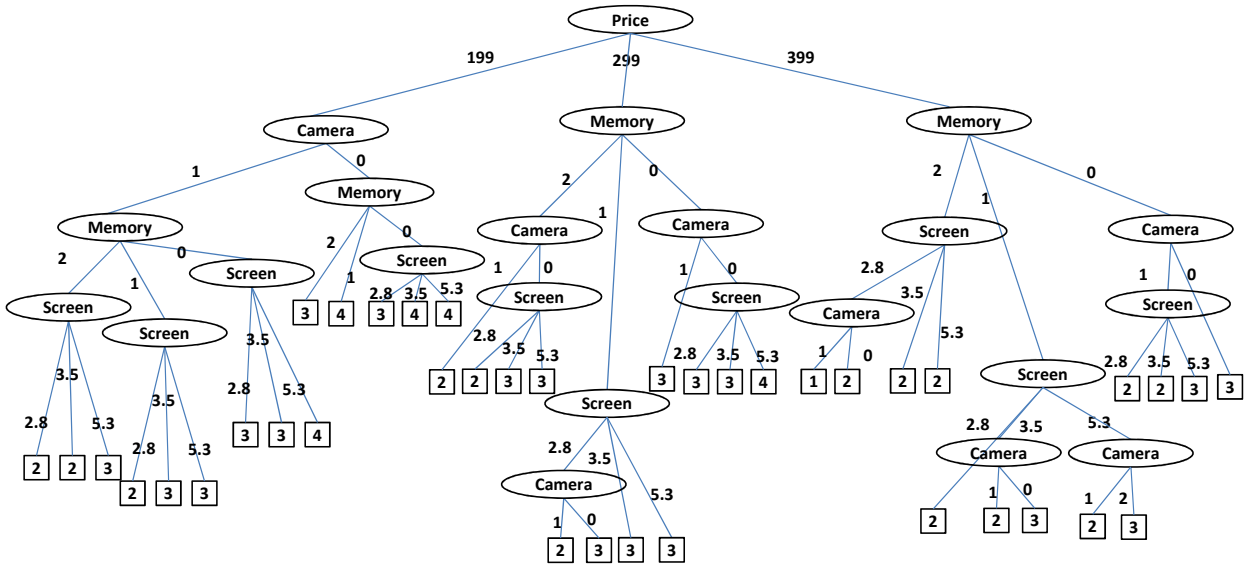


Figure 3.5: Decision tree for new product at $t^{prelife}$ or t^{10}

The decision rules in Figure 3.5 and 3.6 allow estimation of the market share of a specific product. Suppose that the potential competing products are known as shown in Table 3.6. Then, the decision rules can calculate the utility of each competing product, which in turn enables to use Equations (3.1) and (3.2) for market share estimation.

3.3.3 Optimal Life Cycle Design

Figures 3.5 and 3.6 provide different rules for utility estimation. In other words, the market demands at the pre-life and end-of-life stages are different from each other. For example, a smart-phone with a \$199 (for reman \$199.5) price, 3.5-inch screen, 64-GB memory, and 8-MP camera would generate utility value 3 for new product and 2 for reman product. This implies that product design optimized based on the pre-life data only would not be optimal from the end-of-life perspective. To find an optimal product design, the optimal life cycle design model was applied.

In addition to the assumptions in Tables 3.4 through 3.6, the following assumptions were made. The cost of reverse

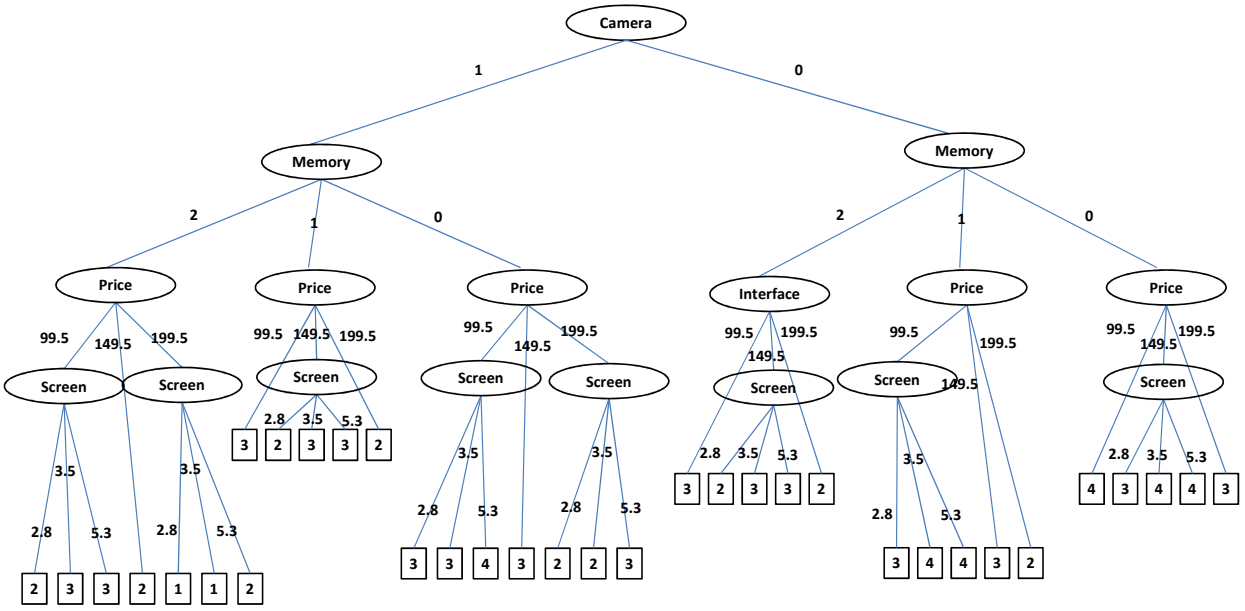


Figure 3.6: Decision tree for reman product at t^{eol} or t^{12}

logistics, sorting, and data scrubbing is \$2 in total and the cost of forward logistics is \$1. The size of new market is 100,000 in terms of the total number of buyers, and the size of reman market is 50% of the new market. As shown in Table 3.5, the remanufacturability, or, reusability rate of a phone is less than 100%, which means that not all the new products can be remanufactured due to functional damages or poor conditions. The take-back loss parameter is one, and only working phones with good conditions would be remanufactured while the remainder is recycled. The recycling profit is \$0.621 [109] and the recycling cost is \$0.39 per cell phone [110]. Lastly, to discount future profit from the end-of-life stage, an annual interest rate of 3% is assumed.

To solve the optimization problem, the Excel risk solver platform was used. Table 3.7 shows the optimization

Table 3.6: Assumptions about competitors information

	High spec product					Mid spec product					Low spec product				
Attributes	New price	Reman price	Screen	Memory	Camera	New price	Reman price	Screen	Memory	Camera	New price	Reman price	Screen	Memory	Camera
	Y_{13}	Y_{23}	X_{13}	X_{23}	X_{32}	Y_{12}	Y_{22}	X_{12}	X_{22}	X_{31}	Y_{11}	Y_{21}	X_{11}	X_{21}	X_{31}
New Utility	3					2					2				
Reman Utility	3					3					2				

results (Column PLCD). To maximize the total life cycle profit, a smart-phone should be equipped with 2.8-inch screen, 64-GB memory, and 16-MP camera. The optimal selling price of the product is $399 at the pre-life stage; the optimal selling price of the remanufactured version is $149.5. The optimal solution also provides optimal production and recovery strategies. The quantity of new products to produce should be 36,552 units; after one year, 26,722 units should be remanufactured, and the rest (9,830 units) recycled. The maximum total profit results in $11,703,000 (in terms of the value at t^{second}).

3.3.4 Discussion

Many traditional design approaches have been focused on maximizing the profit from the pre-life stage only. The PLCD framework with the DTM algorithm is different from them in that it considers the entire life cycle of a product and maximizes the total profit from the life cycle. To demonstrate the benefit of the PLCD framework, this section compares the optimization result of PLCD with those of traditional design approaches. To be specific, two traditional approaches are considered in this section, i.e., pre-life design without any end-of-life recovery and pre-life design with end-of-life recovery. Both approaches seek an optimal product design which maximizes the profit from the pre-life stage; they do not consider how their decision will affect the end-of-life stage. In the latter approach, however, the OEM conducts recovery at the end-of-life stage and tries to maximize the profit from recovery with additional optimization. The additional optimization means optimizing the production quantity and price of the reman product with pre-determined design attributes.

Table 3.7 shows the optimal design and the maximum profit from the traditional approaches. When the pre-life design is conducted, the product is optimized solely for the new market, and different attributes are chosen as the optimal: 3.5-inch screen, 32-GB memory, 16-MP camera. The maximum profit that can be achieved by this design is $10,490,000; if the company conducts recovery at the end-of-life stage (i.e., pre-life design with end-of-life recovery), the profit is increased by $67,000 to $10,556,000. Compared to PLCD, the pre-life designs bring a greater profit at the pre-life stage. However, the benefit of PLCD is revealed when the life cycle profit is considered. In Table 3.7, the profit from PLCD is 10.9% higher than that of the pre-life design with end-of-life recovery.

Previously, the size of the reman market was assumed to be half the size of the new market or $MS^{reman} = 0.5 * MS^{new}$. However, as reported by [85] and [89], the reman market is expected to grow more in the future. To see the effect of an increasing size of reman market and validate the outcome in Table 3.7, a sensitivity analysis is conducted. In Figure 3.7, β denotes the ratio of MS^{reman} to MS^{new}. For both PLCD and pre-life design (with recovery) models, the sensitivity analysis examined how the maximum achievable profit changes as β increases. A different selection of design attributes and consequent demands and amounts of remanufacturable products (D^{reman} and A) are attributed for different gaps in the graph. If β=0, there are no market or demands for the reman products, and no remanufacturing

Table 3.7: Comparative result between PLCD and pre-life design

		PLCD	Pre-life Design	Pre-life Design (+ end-of-life later)
Total profit [$]		11,703,000	10,490,000	10,557,000
Profit for pre-life [$]		10,344,000	10,490,000	10,490,000
Profit for end-of-life [$]		1,359,000	·	67,000
Product attributes	New product price [$]	399	399	399
	Reman product price [$]	149.5	·	99.5
	Screen Size [inch]	2.8	3.5	3.5
	Memory [GB]	64	32	32
	Camera Pixel [MP]	16	16	16
Quantity of Reman product [EA]		26,722	0	26,722
Quantity of Recycled product [EA]		9,830	0	9,830
New product utility		3	3	3
Reman product utility		4	·	4

is conducted; if $\beta = 1$, the size of the reman market is the same as the new market. When β=0, the optimizer will determine the optimal design attributes only from the pre-life stage for both models, which will generate the same design attributes with the total profit of \$8,300,000. When $\beta > 0$, it is expected that the total profit from the PLCD framework is greater than that of the pre-life model except the case of selecting the same design attributes. The results in Figure 3.7 show that both models choose all different designs when $\beta > 0$. When $\beta = 0.6$, the slops of the both models are changed since the upper bound is changed from D^{reman} to A (Equation (13) and (17)). When $\beta = 0.7$, both models select different designs from the previous ones. For $\beta = 0.8$ and $\beta = 0.9$, the upper bounds are changed again, and finally when $\beta = 1$, the optimal design is changed for PLCD. In the illustration example, when $\beta = 0.9$, the profit difference is maximized. The results reaffirm that the PLCD framework with the DTM algorithm is always better than the traditional pre-life design, although the magnitude of the benefit changes depending upon β.

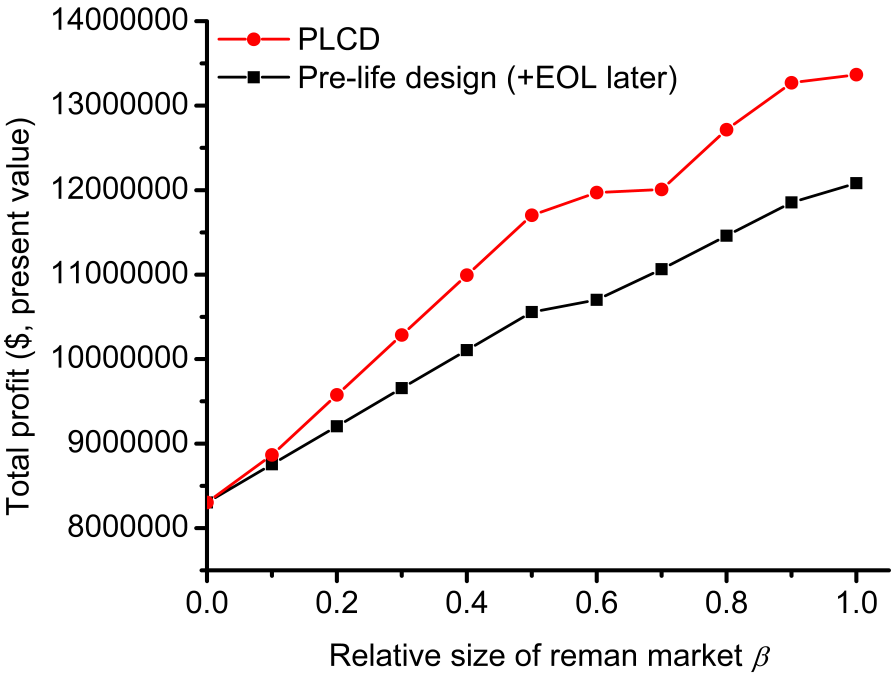


Figure 3.7: Sensitivity analysis of reman market size ratio

3.4 Conclusion

This chapter proposes a new demand modeling technique, demand trend mining (DTM), for product design analytics. The first contribution is the development of the DTM algorithm. In order to capture hidden and upcoming trends of demand, the algorithm combines three different models: decision tree for large-scale data, discrete choice analysis for demand modeling, and automatic time series forecasting for trend analysis. The DTM algorithm dynamically reveals design attribute patterns that affect demands. The second contribution is the new design framework, predictive life cycle design (PLCD), which connects DTM and data-driven product design. The optimization-based model enables a company to optimize its product design by considering the pre-life and end-of-life stages of a product simultaneously. The DTM algorithm interacts with the optimization-based model to maximize the total profit of a product. The smartphone case study demonstrated that there is a hidden source of opportunity for profit and the PLCD framework can help utilize this opportunity. Moreover, the sensitivity analysis reaffirmed that the life cycle design is more preferable than the traditional design method.

The current PLCD framework considers/optimizes two consecutive life cycles of a single product. The model can be extended to accommodate multiple life cycles and multiple products. The current DTM algorithm allows discrete attributes and class variables only, which should be extended to process continuous attributes and class variables. Also,

in reality, it is possible that a product evolves with new attributes. It is important to find out a way how to incorporate emerging attributes into DTM. Text mining [111, 112] and sentiment mining [113] techniques in the domain of product design can be candidates for the management of dynamic attribute sets. On-line review data is a promising source that can provide not only customer preferences but also important emerging attributes. It should be noted that in terms of the performance of predictive models, since Tucker and Kim [14] showed that the predictive model from the preference trend mining outperforms that of the static data mining, the prediction accuracy of the DTM algorithm was not tested in this chapter.

The next chapter will discuss some limitations of discrete preference trend mining and how to deal with them. Also, the newly developed algorithm will be intensively tested with various data sets to validate the performance of its predictability. In terms of data, more commonly available transactional data will be utilized.

Chapter 4

Continuous Preference Trend Mining for Optimal Product Design with Multiple Profit Cycles

In this chapter[1], the Continuous Preference Trend Mining (CPTM) algorithm is proposed to address some fundamental challenges in the context of product and design analytics. The first contribution is the development of a new predictive trend mining technique that captures a hidden trend of customer purchase patterns from accumulated transactional data. Unlike traditional, static data mining algorithms, CPTM does not assume stationarity, but dynamically extracts valuable knowledge from customers over time. By generating trend embedded future data, the CPTM algorithm not only shows higher prediction accuracy in comparison with well-known static models, but also provides essential properties that could not be achieved with previously proposed models: utilizing historical data selectively, avoiding an over-fitting problem, identifying performance information of a constructed model, and allowing a numeric prediction. The second contribution is the formulation of the optimal design problem which can reveal an opportunity for multiple profit cycles. This mathematical formulation enables design engineers to optimize product design over multiple life cycles while reflecting customer preferences and technological obsolescence using the CPTM algorithm. For illustration, the developed framework is applied to an example of tablet PC design in leasing market and the result shows that the determination of optimal design is achieved over multiple life cycles.

4.1 Introduction

Data mining in the context of product and design analytics was suggested as an alternative for knowledge extraction [116]. Traditionally, there are a few methods for capturing customer requirements and preferences such as quality function deployment, conjoint analysis, and discrete choice analysis. These methods resort to direct or close interactions with target customers and generate stated preference data. The strength of using data mining models is to utilize revealed preference data or accumulated data sets related to customers' actual behavior (e.g., transactional data, sales, and on-line reviews) that usually have characteristics of large volume, unstructured form, and timeliness.

Predictive trend mining is a new and emerging data mining area, which is also known as change mining [10, 11] or learning concept drift [12]. Unlike traditional static data mining models with the assumption of stationarity, predictive

[1] Presented in [114] and published in [115].

trend mining is a dynamic and adaptive model that captures trend or change of customer preferences over time.

A tree based data mining algorithm with predictability was used in predictive trend mining. Tucker and Kim [14] proposed the *Discrete* Preference Trend Mining (DPTM) algorithm and suggested a classification of attributes as standard, non-standard and obsolete with respect to a class variable for guiding design engineers. The attributes or features are also known as independent and explanatory variables, and the class variable is a dependent and response variable. It should be noted that due to the fact that the algorithm was developed to deal with discrete class variables and attributes for product portfolio concept generation, the term *Discrete* is added to the original name PTM. For example, five discrete prices {\$99, \$149, \$179, \$199, \$249} were used as the class variable and no design problem was provided. Chapter 3 discussed how to extend the work and proposed that the predictive trend mining technique called Demand Trend Mining (DTM) can benefit optimal life cycle design problems. Utility was used as the discrete class variable and discrete choice analysis was utilized to calculate expected market shares. However, the nature of optimal design problems often requires continuous variables, e.g., price, cost, demand, etc. and discrete class variables might limit the application of design problems. In order to allow continuous variables while capturing a trend, a new method, Continuous Preference Trend Mining (CPTM), is presented in this chapter.

The CPTM algorithm as the method of predictive design analytics will shed light on the initial design problem which has an opportunity for multiple profit cycles. If there are multiple recovery chances for end-of-life products in the near future, a trend of customer preferences and technological obsolescence will be traced and captured at the target time for the optimal initial design. The captured information will then be merged with a product design problem.

Design for multiple life cycles is a design paradigm that enables design engineers to close the loop of a product life cycle and to manage its multiple life cycles. Leasing or sales of service is a representative example of the management of multiple life cycles as shown in Figure 4.1. After designing and manufacturing a product, a lessor (a person possessing a product that is being leased) would lease the product to a lessee (user of the product being leased). At the end of the lease contract or the usage stage, the lessor would take back the product and determine a proper recovery option. If it is profitable, the lessor would lease the product again for a multiple periods of time. Eventually, a product would generate multiple profit cycles, k. Many studies showed that the initial design of a product would determine 70 $\sim$ 85% of total life cycle cost and environmental impact [25, 26, 27], so the selection of initial design attributes is the focus in this chapter, especially from the economic perspective.

In order to combine the design problem with the method of predictive design analytics, design for multiple life cycles is proposed to be formulated as an optimization problem. The formulation determines the design attributes that maximize the total life cycle profit and generate multiple profit cycles. Only a few studies [117, 80] provided mathematical models that realize the total profit from both pre-life (design and manufacturing) and end-of-life stages. Design for multiple life cycles will extend these studies.

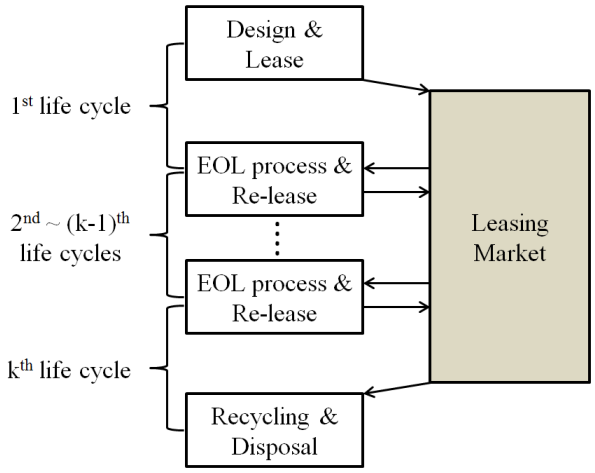


Figure 4.1: Product life cycle in leasing market

As a method of predictive design analytics, the CPTM algorithm is developed to take large sets of transactional data and extract valuable knowledge of customer purchase patterns. The architecture of CPTM will help to predict the target class variable that reflects trend of customer preferences and technological obsolescence over time. By merging the continuous, predictive trend mining technique with an optimization model, the proposed framework will produce an optimal product design that maximizes a total unit profit and eventually reveals an opportunity for multiple profit cycles.

The rest of this chapter is organized as follows. In Section 4.2, the entire methodology is explored with the CPTM algorithm and an optimal product design model for multiple profit cycles. Section 4.3 presents performance tests of CPTM with various data sets. An illustration example of tablet PC design is provided in Section 4.4, and the conclusion and future research directions are presented in Section 4.5.

4.2 Methodology

The entire framework is divided into two phases. Phase 1 is to implement the CPTM algorithm, which entails data preprocessing, trend embedded future data generation, and model tree induction as shown in Figure 4.2. Phase 2 involves an optimal product design for multiple profit cycles by combining the predictive models built from CPTM.

The schematic of the CPTM algorithm shown in Figure 4.3 constructs a predictive model (model tree in Section 2.1.3) at time $n+h$ or h periods ahead based on the historical data sets from time 1 to n. The core part of the algorithm is the generation of trend embedded data. Geometric sampling is developed to capture the trend of the relationship between design attributes and class variables by sampling normalized historical data selectively (i.e., ① and ②).

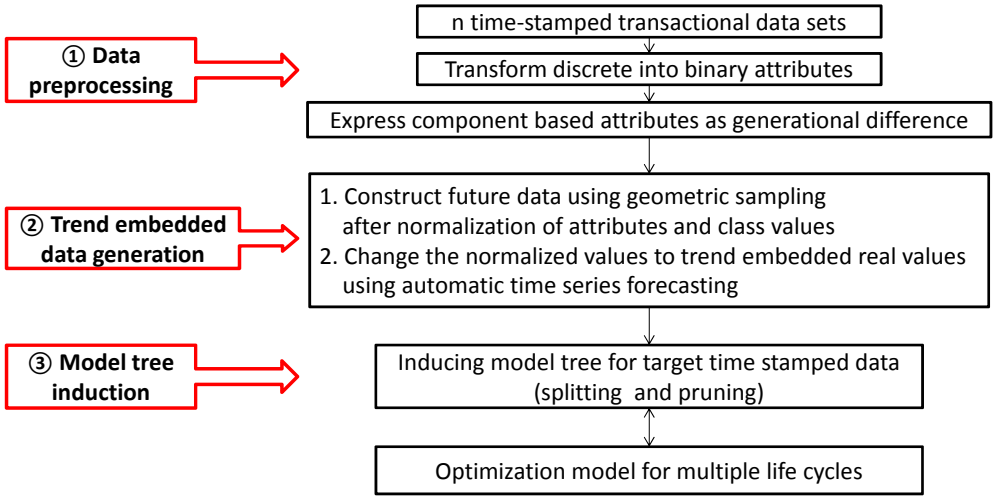


Figure 4.2: Overall flow of methodology

Automatic time series forecasting proposed by Hyndman et al.[96] is used to predict future values of design attributes and class variables. By applying the predicted values to the normalized sampled data, unseen future data at time $n+h$ or D^{n+h} can be generated (i.e., ③). Finally, the future model tree or MT^{n+h} can be built based on the trend embedded data (i.e., ④). The two dotted boxes represent the predicted data and model, which are not available initially.

Discrete PTM, on the other hand, builds a predictive model (decision tree [100]) based on predicted values of Gain Ratio, one of splitting measures. The mathematical form of the Gain Ratio is defined as [118, 14]

$$\text{Gain Ratio(X)} = \frac{Entropy(T) - Entropy_x(T)}{-\sum_{j=1}^{n} \frac{|T_j|}{|T|} \cdot log_2 \frac{|T_j|}{|T|}} = \frac{-\sum_{i=1}^{k} p(c_i) \cdot log_2 p(c_i)[bits] - \sum_{j=1}^{n} \frac{|T_j|}{|T|} \cdot Entropy(T_j)}{-\sum_{j=1}^{n} \frac{|T_j|}{|T|} \cdot log_2 \frac{|T_j|}{|T|}} \tag{4.1}$$

where X is a set of attributes, T is a data set, and T_j is a subset of the data T after splitting. The denominator represents the information generated by splitting the data set T into n partitions. The numerator represents the amount of uncertainty reduction by splitting on attribute x. Entropy quantifies the expected value of the information in bits. $p(c_i)$ represents the probability mass function of a class variable c_i and k is the number of class values.

The concept of building a tree model based on the predicted Gain Ratio was initially proposed by Böttcher and Spott [10]. The predicted Gain Ratio provides a way to build a future classification tree without real data but there are some disadvantages. First, there is a strong possibility of over-fitting since no pruning process is suggested. Highly branching trees risk over-fitting the training data and performing poorly on new samples. Pruning can help to determine the optimal size of trees. Second, no performance result of built models can be estimated since there is no test data. Third, the Gain Ratio based methods are only applicable to classification models or discrete class variables. It will be

shown that by generating the target data, the CPTM algorithm can provide a way to utilize historical data selectively, avoid an over-fitting problem, and identify performance information of constructed model. More importantly, CPTM adopts a tree induction model that allows the use of continuous class variables.

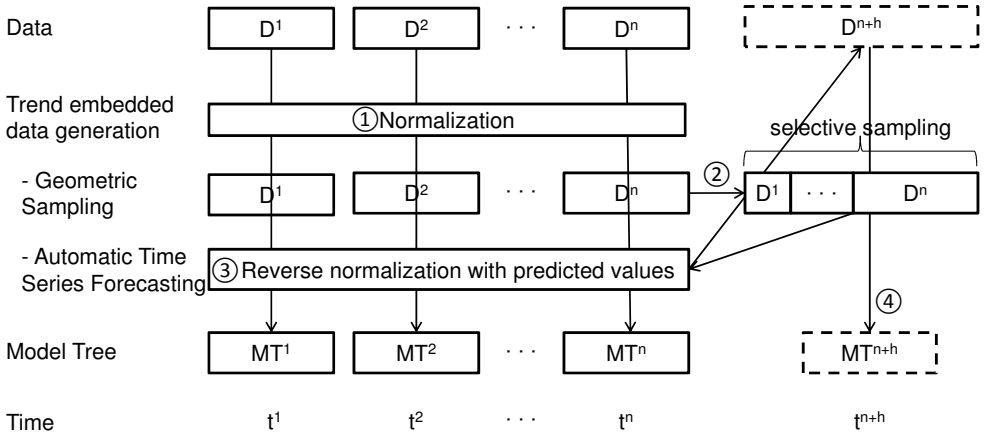


Figure 4.3: A schematic of CPTM algorithm

Usually the process of data mining consists of data collection and selection, cleaning and transformation, pattern discovery, and interpretation. In the product design domain, text and web mining [119] provides a way for design engineers to collect and analyze customer preference data (e.g., review data), including identifying product attributes and modeling customer ratings [120, 121, 122, 123, 124]. In this study, our focus is limited to the pattern discovery and interpretation stage.

4.2.1 Phase 1: Continuous Preference Trend Mining

Data Preprocessing

The first step, data preprocessing, is a data preparation technique for trend mining. It starts by gathering and organizing *n*-time stamped transactional data sets. An example of a data set is shown in Table 6.4. The data set consists of a set of attributes and one class variable. In the example, there are 8 different attributes of a product and a class variable, price, which customers paid in their transactions. Even though any class variables that researchers are interested in can be selected, paid price or market value was used in this study since it is directly related to customer preferences. Sales or demand can be another candidate. There is no restriction on the data except that the class variable should be continuous. Both discrete and continuous attributes can be dealt with by using the proposed approach. In this study, only one class variable is modeled. In order to allow more than one class variable, a multivariate tree [125] can be used instead of a univariate tree (i.e., model tree).

Next, discrete attributes are transformed to a set of binary attributes. In the case of attributes with significant improvement in their values (which are component based attributes, e.g., Hard drive, CPU, etc.), the values need to be expressed as a *generational difference* [99]. The generational difference is a relative scale that can be acquired by comparing the generational gap between the target part and the latest cutting-edge part which corresponds to the minimum generation or zero. As time passes, a new part is introduced in the market and the generational difference of the existing part is increased. We assume that customers perceive the relative generational gap of components with a given time, and a company has expected values or a roadmap of the components in the near future. The generational difference is utilized to represent the technological obsolescence and its effect over time. In Table 4.10, an example of generational difference is shown over time and the cutting-edge part has a value of zero.

Before moving to the next step, it can be checked whether structural changes or trends are in the data. In this study, two possible trends are identified. First, levels under each attribute and class variable can have increasing, decreasing or cyclical patterns. For example, the display size of cell phones can have an increasing pattern. In order to detect this kind of trend, it is useful to visualize data. Statistically, Spearman's rho test of trend and Mann-Kendalls tau test of trend are available [126]. Second, there are some trends in terms of relationship between design attributes and class variables over time. For example, the memory size of notebooks might be an important factor for the purchase of the products a couple of years ago but some technological advances can change the importance of the memory size in the next year. There is no known method to detect this kind of trend but one possible way is to apply the tests of trend to the coefficients of regression models. If both trends are not detected (i.e., static case), CPTM will generate the same result with the simple model tree, and other static models (e.g., regression, neural network, SVM, etc.) can be applied to the latest data set or the entire data set depending on the characteristics of data.

Trend Embedded Future Data Generation

The second step is the generation of trend embedded future data. In the previous section, two different trends were introduced in data. The automatic time series forecasting is the technique that captures the first type of trend, and the geometric sampling that is newly proposed in this study helps to capture an underlying relationship between design attributes and a class variable (i.e., the second type of trend) by selectively utilizing historical data.

When there are a series of time stamped data points, $\{x_1, x_2, \ldots, x_n\}$, where x_t stands for a data point at time t, a couple of different techniques can be applied to forecast a data point at $n+1$ or one-step-ahead forecast. As a heuristic, it is possible to take either the latest data point or the average of all historical data for the forecast. Simple moving average is a method to smooth a time series over last k observations though the selection of k can be a heuristic. Exponential smoothing is one of well-known time series analysis methods, and the simplest form is given by Equation (4.2) [96].

$$\hat{x}_{n+1} = \lambda x_n + (1-\lambda)\hat{x}_n = \lambda x_n + \lambda(1-\lambda)x_{n-1} + \lambda(1-\lambda)^2 x_{n-2} + \cdots + \lambda(1-\lambda)^3 x_{n-3} + \lambda(1-\lambda)^{n-1} x_1 + (1-\lambda)^n \hat{x}_1 \tag{4.2}$$

where $\hat{x}_t$ is a forecast at time t and λ is a constant between 0 and 1. The exponential smoothing is a weighted moving average of all time series with exponentially decreasing weights defined by λ. The expanded form shows that recent values have a greater weight than old ones. A total of 30 exponential smoothing models are classified based on the combination of trend, seasonal, and error components. There are two error components (additive and multiplicative), three seasonal components (none, additive, and multiplicative), and five trend components (none, additive, multiplicative, additive-damped, and multiplicative-damped). For example, a model with all additive components can be expressed as (trend+seasonal+error) and a model with all multiplicative components is (trend$\times$seasonal$\times$error). Hyndman et al. [96] provided all the classifications. We adopted the automatic forecasting method [127]. First, apply all the 30 exponential smoothing models and estimate initial states and parameters using maximum likelihood estimation. Second, choose the best model according to one of the following criteria: Akaike's information criterion (AIC), corrected Akaike's information criterion (AICc) or Bayesian information criterion (BIC) [127].

After the geometric sampling process which will be introduced shortly, the sampled normalized data set for the target time is finally transformed to the real value data by applying predicted values of each attribute and class variable. The minimum and maximum values of each attribute and class at the target time are predicted (i.e., the first type of trend) by the automatic time series forecasting algorithm. By adopting the automatic algorithm, users do not need to resort to their own knowledge for models and parameters.

The second type of trend cannot be captured by time series analysis methods since the underlying relationship between design attributes and class variables is hidden. However, similar to the exponential smoothing, required traits include dynamically utilizing all past observations and applying decreasing weights in order to reflect underlying trends of the relationship between design attributes and class variables. Previously proposed trend mining models [10, 14] did not consider the dynamics of relative importance of historical data. For example, Böttcher and Spott [10] used a polynomial regression method to predict the future Gain Ratio. This implicitly gave equal weights for all historical data. If historical data is available and older data sets contain more errors (this can be viewed as outliers), the accuracy of the predictive model will be diminished. The CPTM algorithm, on the other hand, provides a dynamic selection of historical data for the reflection of upcoming hidden trends by assigning exponentially decreasing weights to old data sets.

The geometric sampling is a method to sample historical data selectively for the second type of trend. Before sampling, each attribute and class variable should be normalized within a single time step. The tth term of the geo-

metric sampling or a_t which gives the number of instances (data points) that needs to be sampled at time t is given by Equation (4.3).

$$a_t = a(1-\alpha)^{n-t} \tag{4.3}$$

where a is an original number of instances, (1- α) is a common ratio in geometric series, α is a smoothing factor ($0 \leq \alpha \leq 1$) and n is the latest time. The smoothing factor α can be considered a characteristic of product domain in terms of relationship between design attributes and class variable. Table 4.1 indicates that when α is close to 1, only the latest data set is useful, and the product domain is technology sensitive and rapidly changing. When α is close to 0, all data sets are valid for future target time and the product domain has a quite insensitive and slowly changing characteristic.

Table 4.1: α value and product domain

α value	sampling	product domain
≈ 1	only the latest data set	technology sensitive, drastically changing
≈ 0	all data sets	insensitive and slowly changing

In the geometric sampling, α is defined as a smoothing factor to generate $t = n$ data using $t = 1$ to $t = n-1$ data when $t = 1$ to $t = n$ data are available. α is obtained by Equation (4.4).

$$\arg\min_{\alpha} E \tag{4.4}$$

where E is a performance measure (e.g., error metrics such as mean absolute error, root mean-squared error, and relative squared error, etc.) tested for a model tree constructed from $t = 1$ to $t = n-1$ data sets (as training data) with $t = n$ data (as test data). The data for building a model tree can be sampled by Equation (4.3). A model tree will be introduced in the next section.

For example, if $t = 1$ to $t = 10$ normalized data sets are available, using $t = 1$ to $t = 9$ data sets, a model tree can be constructed with different α values and predicted values of attributes and class variables at $t = 10$, and validated with $t = 10$ data. Table 4.2 shows the best α example in terms of the performance measure, mean absolute error which is the average deviation between predicted and observed class variable price, with simulated data sets (each has a thousand instances or $a = 1000$). For $\alpha = 0.9$, the number of total instances (1111) comes from a thousand instances ($1000(1-0.9)^0$) from $t = 9$ data, a hundred instances ($1000(1-0.9)^1$) from $t = 8$ data, ten instances ($1000(1-0.9)^2$) from $t = 7$ data, and one instance ($1000(1-0.9)^3$) from $t = 6$ data based on Equation (4.3). The required numbers are sampled randomly using a random number generator. The sampled normalized data becomes real value data after applying predicted values of attributes and class variables. Then, a model tree can be built based on this data and tested

with $t = 10$ data. The best α is 0.4 based on the performance measure, mean absolute error in Table 4.2.

Table 4.2: Example of best α selection

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Mean Absolute Error	39.6	37.4	35.4	34.8	32.8	33.0	33.4	33.7	34.7	35.5	37.0
Total number of instances	9000	6125	4321	3199	2477	1998	1666	1427	1250	1111	1000

Based on the selected α, a number of required instances for each data set is determined and sampled as in the example but using $t = 1$ to $t = 10$ data sets this time. By applying predicted values of attributes and class variables at a target time to the sampled normalized data, trend embedded future data is finally generated at the target time. Table 4.2 shows that the total number of samples can be varied depending on the selected α. Based on the smoothing factor, only the latest data or all data can be used in the extreme case.

A graphical example of the trend embedded future data generation is depicted in Figure 4.4. The values of original data sets are normalized within a single time step and then sampled using the geometric sampling method with the selected α. In the example, suppose that the first and the last instances were sampled from the $t = 10$ data set. By applying predicted minimum and maximum values from the time series prediction technique, real values are predicted at the target time. For example, the display size is getting bigger, and the generated target data set reflects the trend (e.g., refer to the small arrows).

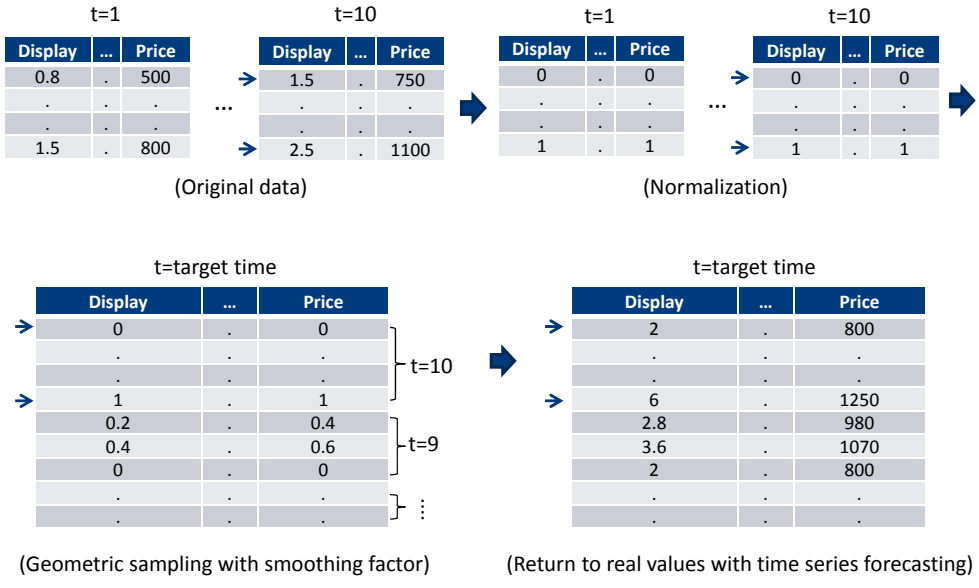


Figure 4.4: Graphical example of trend embedded data generation

By generating future data, two advantages can be achieved. First, performance information of built models can be provided similar to normal data mining processes. The predicted Gain Ratio based models in Section 4.2.1 cannot give

test data but the generated data from CPTM can work as test and validation data. The 10-fold cross-validation technique [119] is a popular way to get the performance (i.e., prediction of class variables) information when a validation data is not available. In the 10-fold cross-validation, the generated data is randomly partitioned into 10 subsamples and validation processes are repeated 10 times. Each time a model is built using 9 subsamples and validated with one remaining subsample. Then, an average performance error can be estimated. Second, pruning can be implemented based on the generated data to reduce the risk of over-fitting. The predicted Gain Ratio based models classify class values without data so that no comparison can be made between a node and subtree for the pruning. In the next section, pruning in the model tree algorithm will be introduced.

Model Tree Induction

The third step is to build a model based on the newly generated data set from the second step. In this step, the knowledge and hidden patterns between the new values of attributes and class variables are mined using a model tree. The result of the model tree is a piecewise linear regression equation depending on a given data set, which can approximate non-linear functions. Figure 4.5 shows an example of a model tree. The model tree gives three different linear models to express the non-linearity with two attributes: A and B. On the other hand, a decision tree that was used in the other trend mining algorithms classifies discrete or categorical class variables.

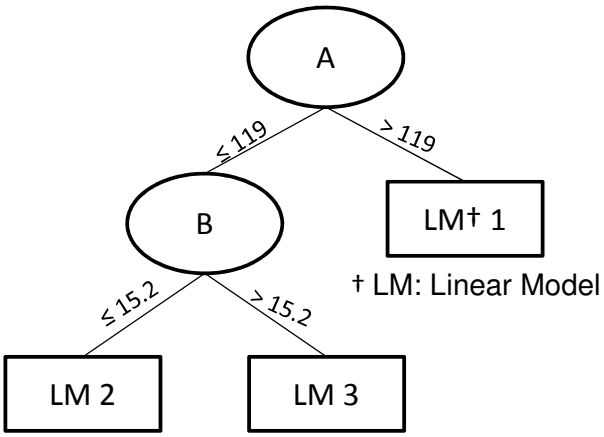


Figure 4.5: Example of model tree

The M5 model tree was initially proposed by Quinlan [128]. After comprehensive descriptions of model tree induction including a pseudocode by Wang and Witten [129], the model tree has received attention from researchers. Wang and Witten's model tree algorithm is known as M5P. The basic operation is splitting, and the splitting is based on standard deviation reduction (SDR) in Equation (4.5) [129].

$$\text{SDR} = stdev(T) - \sum_i \frac{|T_i|}{|T|} \times stdev(T_i) \tag{4.5}$$

where stdev() is a standard deviation, $|\cdot|$ stands for the number of instances, T is all instances, T_1, T_2, . . . are result sets from splitting attributes. An attribute is determined as a node when it has a maximum SDR compared to all other attributes' SDR. If no attribute can reduce a standard deviation of class values, a model tree will be identical to a simple linear regression model. For example, there are two attributes and one class variable in the model tree example in Figure 4.5. stdev(T) is the standard deviation of the class values. All possible split points of the two attributes are used to estimate SDRs of the class values after splitting. Then, the one with the maximum SDR becomes the split point and the attribute of the split point becomes the node. The termination criterion of splitting in the M5P is when the number of instances is less than four or when the standard deviation at a node is less than 0.05*stdev(T). Once the splitting operation is finished, instances at the leaf nodes are used to build linear models.

A pruning procedure can reduce size of a tree and the risk of over-fitting. The M5P algorithm uses post-pruning or backward pruning, which means the pruning process starts after a tree reaches a leaf node. If the lower estimated error is expected when errors in non-leaf nodes and subtree are compared, the subtrees are pruned to be leaves. The expected error of subtrees is the weighted average of each node's error by the proportion of sample sizes, and the expected error of non-leaf nodes is in Equation (4.6) [129].

$$\frac{n+v}{n-v} \frac{\sum_{\text{instances}} |\text{deviation from predicted class value}|}{n} \tag{4.6}$$

where n is the number of instances at the non-leaf node and v is the number of parameters in a linear regression model in the node. The second fraction represents the average of absolute difference between the predicted value and the actual class value over each of instances that reach the node. The first fraction is the compensation factor to simplify the regression model in the node.

The manual implementation of the model tree in Figure 4.5 is provided to show how it works. Based on the sample data in Table 4.3, the model tree in Figure 4.5 is manually built. The sample data has two attributes, A and B, and one class variable C.

Table 4.3: Sample data for model tree

A	B	C
200	14.5	10
140	20	26
90	14.4	29
98	13.5	32
86	16	34
50	24	44

Table 4.4 shows all standard deviation reduction (SDR) calculations for determining the root node and splitting point of the model tree. First, the class variable and each attribute are grouped, and values of the attribute are sorted from smallest to largest. For each mid-point, calculate the standard deviation of class values in the divided groups. For example, with the mid-point 88 (the second row), $stdev(T1)$ and $stdev(T2)$ represent the standard deviation of $\{44, 34\}$ and $\{29, 32, 26, 10\}$. $stdev(T)$ represents the standard deviation of all the class values. Then, the final column SDR is calculated based on Equation (4.5), and the mid-point that produces the maximum SDR is the splitting point (i.e., 119 of attribute A).

Table 4.4: Determining a root node of model tree

C	A	mid-point	stdev(T)	stdev(T1)	stdev(T2)	SDR
44	50	68	11.2	N/A	9.5	N/A
34	86	88	11.2	7.1	9.8	2.3
29	90	94	11.2	7.6	11.4	1.7
32	98	119	11.2	6.5	11.3	3.1
26	140	170	11.2	6.9	N/A	N/A
10	200	N/A				
C	B	mid-point	stdev(T)	stdev(T1)	stdev(T2)	SDR
32	13.5	14	11.2	N/A	12.4	N/A
29	14.4	14.5	11.2	2.1	14.4	0.9
10	14.5	15.3	11.2	11.9	9.0	0.7
34	16	18	11.2	11	12.7	-0.4
26	20	22	11.2	9.5	N/A	N/A
44	24	N/A				

When the value of attribute A is greater than 119, only two instances reach the node. The termination criterion of the M5P algorithm (i.e., less than four instances) stops further splitting for this branch. For the other branch, there are four instances and the standard deviation at the node (6.5) is greater than the other criterion (i.e., 0.05*stdev(T)=0.56). After removing the instances that are greater than 119, the same procedure can be applied as shown in Table 4.5. In this case, two splitting points (i.e., 88 of attribute A and 15.2 of attribute B) produce the same SDR so that either of them can be selected and the model performance will be the same. Figure 4.5 shows the case that 15.2 of attribute B is selected. All the nodes have less than four instances so that the splitting operation of the model tree is completed. Finally, the instances at the leaf nodes are used for regression models. Due to the small number of instances, all leaf nodes take a simple model i.e., LM1: C=18, LM2: C=30.5, and LM3: C=39.

A pruning procedure compares the expected error of leaf nodes and a non-leaf node. The non-leaf node B has two leaf nodes and their expected error can be calculated as follows: the absolute difference between the predicted and the actual class value is averaged at the each node and weighted by the proportion of sample sizes ($\frac{2}{4}(\frac{|29-30.5|}{2} + \frac{|32-30.5|}{2}) + \frac{2}{4}(\frac{|34-39|}{2} + \frac{|44-39|}{2}) = 3.25$). The internal regression model at the node B (C=1.31*B+12.59) is then used to calculate the expected error (2.05) based on Equation (4.6). Also by dropping the parameter of the internal

Table 4.5: Determining the second node of model tree

C	A	mid-point	stdev(T)	stdev(T1)	stdev(T2)	SDR
44	50	68	6.5	N/A	2.5	N/A
34	86	88	6.5	7.1	2.1	1.9
29	90	94	6.5	7.6	N/A	N/A
32	98	N/A				
C	B	mid-point	stdev(T)	stdev(T1)	stdev(T2)	SDR
32	13.5	14	6.5	N/A	7.6	N/A
29	14.4	15.2	6.5	2.1	7.1	1.9
34	16	20	6.5	2.5	N/A	N/A
44	24	N/A				

regression model (C=34.75), another expected error (4.63) can be calculated but this model can be ignored due to the higher expected error. Since the expected error of the node B is lower than that of the leaf nodes, the tree should be pruned and the internal regression model becomes a leaf node. Similarly, by comparing the node A and leaf nodes, the pruning operation can be determined and it turns out that the tree should be pruned. After the pruning procedure, one regression model (C=-0.21*A+52.21) replaces the three regression models.

The unique contribution of this Phase 1 is to propose a new data generation scheme for a target time, which reflects two different trends. By applying the model tree algorithm to this predicted data set, this section shows some crucial properties that could not be achieved with the previous models [10, 14]: dynamic selection of historical data, avoidance of over-fitting problem, identification of performance information of constructed model, and allowance of a numeric prediction. Section 4.3 will show empirical test results with higher prediction accuracy.

4.2.2 Phase 2: Optimal Product Design for Multiple Profit Cycles

As shown in Figure 4.1, products can have multiple life cycles in the leasing market. When design engineers determine the initial product design over the multiple life cycles, they should consider not only the profit from the initial lease, but also the profit from the recoveries and re-leases, which can be a hidden source of profits. Usually, the latter part is ignored in the initial design stage due to the absence of supporting models. The CPTM results from Section 4.2.1 are expressed as model tree functions and will help to reveal the hidden source of profits.

The optimal product design for multiple profit cycles is formulated as a mathematical model and the overall architecture of the model is depicted in Figure 4.6. Model tree functions are used to reflect customer preferences and technological obsolescence over time. In order to address the reliability of target products over time, a reusability function is formulated, which will give probabilities of reusable and non-reusable products. The probabilities affect the cost of end-of-life processes. While the optimizer evaluates the unit profit of a given set of attributes that are decision variables, model tree and reusability functions will take those decision variables and return the unit price and

the cost of end-of-life process at a given time t respectively.

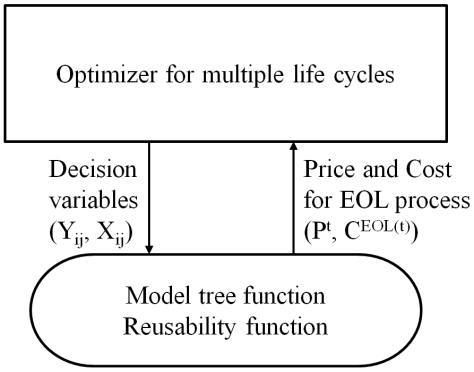


Figure 4.6: Architecture of optimal design with CPTM

Problem Statement

The unit profit of a design over multiple life cycles is obtained by a mathematical model. The model is summarized as the following optimization problem.

Objective

· Maximize unit profit of the product for its life cycle

Constraints

· Uniqueness of design attributes

Decision variables

· Target product design attributes

Given inputs

· Historical transactional data as a set of attributes and paid price

· Generational information of parts

· Reliability information

· Cost of manufacturing and new parts

· Cost of reconditioning and logistics

Mathematical Formulation

Objective Function The objective function is expressed as the summation of unit profits, which is the difference between unit price and unit cost at a given time t in Equation (4.7).

$$Maximize\ f = \sum_{t} \frac{1}{(1+r)^{\delta}}(p^t - c^t) \tag{4.7}$$

Since the multiple life cycles occur in the future, an annual interest rate r should be applied to discount the value. For the present value, $\frac{1}{(1+r)^{\delta}}$ is multiplied and δ is the number of the years.

A unit price at time t is derived from the model tree function $MT^t()$ in Equation (4.8). Binary decision variables, Y_{ij}, represent the level of non-component based attributes such as weight, size, color, etc. and X_{ij} represent the level of component based attributes or replaceable and upgradable attributes such as battery, memory, CPU, etc. A unit cost is divided into three different costs. First, when time t is the starting time (t_1), it is the production of new products, and the unit cost consists of manufacturing costs and forward logistics costs in Equation (4.9). The manufacturing cost is affected by X_{ij}. If product attribute i has the level of j, X_{ij} equals 1; otherwise, it equals 0. Second, when time t is the take-back time, it is the remanufacturing of take-back products, and the unit cost consists of end-of-life process costs, and inverse and forward logistics costs in Equation (4.10). Third, when a product eventually reaches the point that is not profitable (t_{end}), it will have a unit price of recycling and a cost of disposal in Equation (4.11) and (4.12).

$$p^t = MT^t(Y_{ij}, X_{ij}), where\ t \neq t_{end} \tag{4.8}$$

$$c^t = \sum_{j} c_j^{manufacturing} X_{ij} + c^{forwardlogistics}, where\ t = t_1 \tag{4.9}$$

$$c^t = c^{inverselogistics} + c^{EOL(t)} + c^{forwardlogistics}, where\ t = t_{take\text{-}back} \tag{4.10}$$

$$p^t = p^{recycling}, where\ t = t_{end} \tag{4.11}$$

$$c^t = c^{inverselogistics} + c^{forwardlogistics} + c^{disposal}, where\ t = t_{end} \tag{4.12}$$

Constraints Equation (4.13) imposes that each product attribute i has a unique attribute level j. In other words, finding a unique combination of each design attribute is the design problem.

$$\sum_{j} Y_{ij} = 1, Y_{ij} \in (0,1),\ \sum_{j} X_{ij} = 1, X_{ij} \in (0,1) \tag{4.13}$$

A probability of reusable parts β or a reusability function is defined as the multiplication of each part's reliability at

time t in Equation (4.14). Equation (4.15) formulates the cost of end-of-life processes as manufacturing costs with new parts and reconditioning costs with old parts. Reconditioning is conducted with probability β. If a part is not reusable, then a new part should be used. Because a part's manufacturing cost differs by design decisions, remanufacturing with new parts is formulated as a function of X_{ij} with a probability of non-reusable parts (1 - β). Table 4.6 shows the probability of reusable and non-reusable parts at different time t with the assumption that the reliability of a product will be back in a state as new after end-of-life processes. Therefore, memorylessness is satisfied.

$$\beta = \prod_i \sum_j \gamma_j(t) X_{ij} \tag{4.14}$$

$$c^{EOL(t)} = \sum_j c_j^{manufacturing} X_{ij}(1-\beta) + c^{reconditioning}\beta \tag{4.15}$$

Table 4.6: Probability of reusable and non-reusable parts at different time t

Time	$t=1$	$t=2$	$\cdots$	$t=n$
Prob. of non-reusable parts	(1- β)	$(1-\beta)^2$+β(1- β)=(1- β)	$\cdots$	(1- β)
Prob. of reusable parts	β	β^2+(1- β)β=β	$\cdots$	β

The contribution of this Phase 2 is to formulate the optimal product design model with multiple life cycles. In order to address some issues on the multiple life cycles such as customer preferences, technological obsolescence, and reliability over time, model trees from CPTM and reusability functions are combined in the optimization model.

4.3 Performance Test of CPTM

In this section, a set of different data are tested with the CPTM algorithm. In order to understand the mechanism of CPTM, Section 4.3.1 and Section 4.3.2 provide simple data sets. A real data set is also tested in Section 4.3.3 to verify the performance of the CPTM algorithm in a real situation. Section 4.3.4 deals with the most complex data that will be used for the statistical analysis and the illustration study in Section 4.3.5.

Four different static models were compared with the dynamic model, CPTM: linear regression, model tree (M5P), support vector machine (SMOreg), and neural network (Multilayer Perceptron). Weka [107] was used to implement these models, and the names in the parenthesis represent the equivalent algorithms. For the automatic time series forecasting, R [108] was used with the package, *forecast* [127]. All static models construct a predictive model based on the latest data set (*latest* in Table 4.8) or all historical data sets (*all* in Table 4.8) as heuristics. On the other hand, CPTM utilizes all historical data selectively and builds a predictive model based on the generated data set. It is important to realize that the CPTM algorithm also uses the model tree but the difference is in the use of trend

embedded target data.

As a performance measure, mean absolute error (MAE) and root mean-squared error (RMSE) were used [119]. Equation (4.16) and (4.17) show the MAE and the RMSE with the predicted class values, $b_1, b_2, \cdots, b_m$ and the actual class values, $d_1, d_2, \cdots, d_m$.

$$\text{Mean Absolute Error} = \frac{|b_1 - d_1| + \cdots + |b_m - d_m|}{m} \tag{4.16}$$

$$\text{Root Mean Squared Error} = \sqrt{\frac{|b_1 - d_1|^2 + \cdots + |b_m - d_m|^2}{m}} \tag{4.17}$$

4.3.1 Test with Data Generated from Stationary Linear Mapping Function

The data shown in Figure 4.7 was generated by stationary linear mapping functions over time. There are two nominal attributes, A and B, and one class variable, Class. The values of column A increase by two and those of column B by one over time, which represents the first type of trend. In order to generate the class values, the first five instances used a mapping function, $Class = 0.1 * A + 0.9 * B + Random(-0.2 \sim 0.2)$ and the remaining five instances used a mapping function, $Class = 0.4 * A + 0.2 * B + Random(-0.2 \sim 0.2)$ with some randomness in the functions, which represents the second type of trend. Since this is a stationary case, all data sets from $t = 1$ to $t = 8$ have the same mapping functions.

The goal is to construct a predictive model for $t = 8$ data with $t = 1$ to $t = 7$ data sets. First, the values of each attribute and class variable were normalized within a single time step. Second, based on Equation (4.4), the smoothing factor, $\alpha = 0$, was selected using the built model tree from $t = 1$ to $t = 6$ data with different αs on Equation (4.3) and tested with $t = 7$ data in terms of the MAE. Then, the selected α gave the number of samples from each normalized data set based on Equation (4.3). Since $\alpha = 0$, all 70 normalized data were sampled. Third, the automatic time series forecasting was conducted for the original values of attributes A, B, and class variable $Class$ in Table 4.7. By applying the predicted minimum and maximum values to the sampled normalized data, Figure 4.7 shows the resulted trend embedded data set for the target time $t = 8$. Finally, the model tree algorithm was applied to the generated data set and the built model tree is the predictive model from the CPTM algorithm. The model tree was pruned so that 10 linear models were reduced to only two linear models. The pruned model showed almost the same performance accuracy compared to the unpruned tree. Table 4.8 shows the result of the performance test.

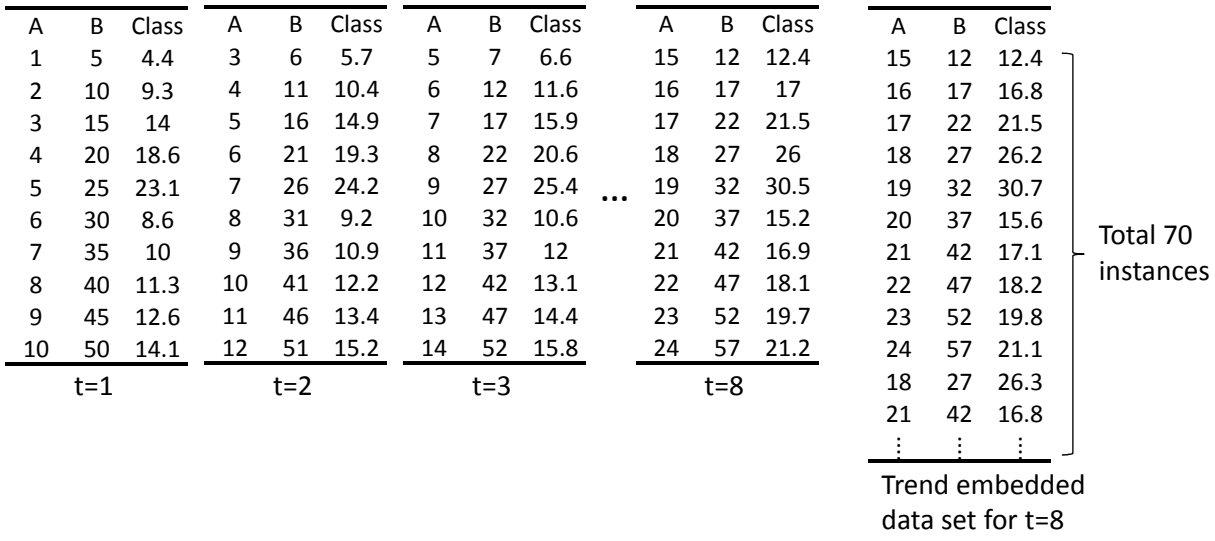


Figure 4.7: Data from stationary linear mapping function and generated future data

4.3.2 Test with Data Generated from Stationary Non-Linear Mapping Function

The data shown in Figure 4.8 was generated by stationary non-linear mapping functions over time. Two nominal attributes, A and B are the same as in Section 4.3.1 but non-linear mapping functions were used: $Class = 0.01 * A^2 + 0.9 * B + Random(-0.2 \sim 0.2)$ for the first five instances and $Class = 0.2 * \sqrt{A} + 0.3 * B + Random(-0.2 \sim 0.2)$ for the last five instances.

The goal is to construct a predictive model for $t = 8$ data with $t = 1$ to $t = 7$ data sets, and Figure 4.8 shows the trend embedded data set for the target time. The smoothing factor, $\alpha = 0.3$, was selected using the built model from $t = 1$ to $t = 6$ data and tested with $t = 7$ data. Also the automatic time series forecasting was conducted in Table 4.7. Based on the smoothing factor, 27 normalized instances were sampled and the predicted values were applied to them. The model tree was pruned so that 8 linear models were reduced to only two linear models. The pruned model showed a little bit higher performance accuracy compared to the unpruned tree. Table 4.8 shows the result of the performance test.

4.3.3 Test with Real Data

Second-hand values or buy-back prices of cell phones [130] were tested with CPTM. Since the data set was obtained with the list of target cell phones, all attribute values were the same but buy-back prices were varied over time. Due to market penetration, the market value of the same products has a tendency to go down over time. After preprocessing the original data, monthly data sets of 155 cell phones from June 2009 to March 2010 were tested with 10 different

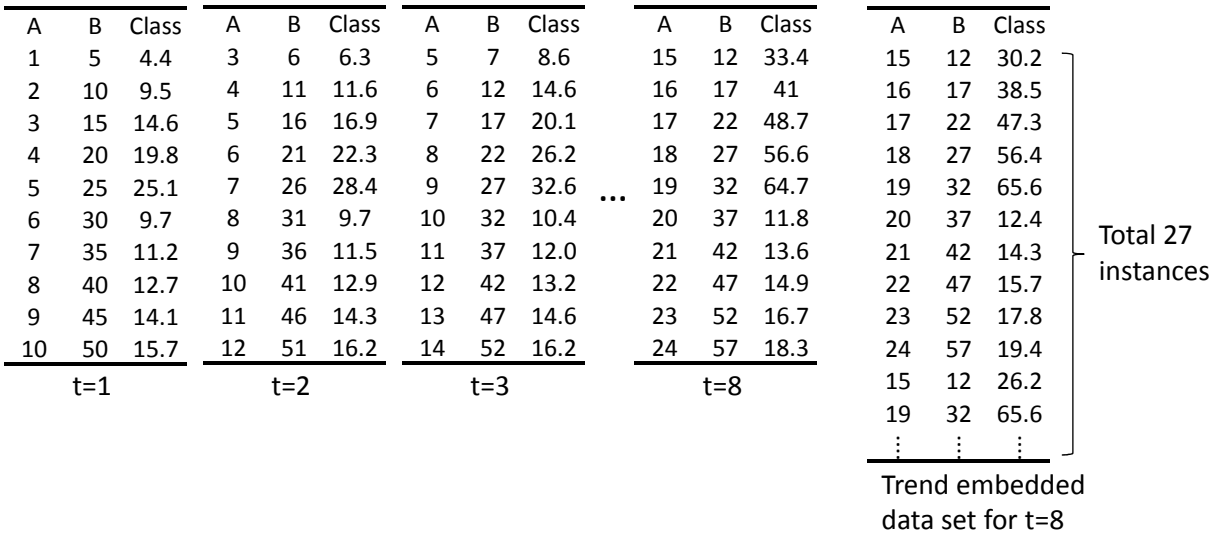


Figure 4.8: Data from stationary non-linear mapping function and generated future data

attributes: camera pixel, talk time, touch screen, weight, memory slot, wiFi, MP3, GPS, bluetooth, and 3G.

The goal is to construct a predictive model for the $t = 10$ data with $t = 1$ to $t = 9$ data sets. The smoothing factor, $\alpha = 0$, was selected using the built model from $t = 1$ to $t = 8$ data and tested with $t = 9$ data. Also the automatic time series forecasting was conducted in Table 4.7. Table 4.8 shows the result of performance test with this real data.

Table 4.7: Forecast results

			t=7 (Latest)	Forecast	t=8 (Target)
Stationary linear data	A	Min	13	15	15
		Max	22	24	24
	B	Min	11	12	12
		Max	56	57	57
	Class	Min	11.3	12.42	12.4
		Max	29.7	30.73	30.5
Stationary nonlinear data	A	Min	13	15	15
		Max	22	24	24
	B	Min	11	12	12
		Max	56	57	57
	Class	Min	11.75	12.37	11.79
		Max	56.9	65.55	64.7
Real data	Class	Min	1	0.9	0
		Max	379	379	395

Table 4.8: Performance results

			MAE	RMSE
Stationary linear data	Dynamic	CPTM	1.30	1.57
	Static (latest/all)	Linear Regression	3.93 / 3.83	5.12 / 5.06
		Model Tree	3.87 / 2.59	5.44 / 4.93
		SVM	3.80 / 3.71	5.12 / 5.31
		Neural Network	3.61 / 5.46	4.99 / 6.89
Stationary nonlinear data	Dynamic	CPTM	6.77	8.88
	Static (latest/all)	Linear Regression	11.80 / 14.64	15.75 / 17.35
		Model Tree	11.86 / 8.65	18.34 / 15.96
		SVM	12.73 / 15.68	18.19 / 20.98
		Neural Network	13.24 / 15.50	19.59 / 20.58
Real data	Dynamic	CPTM	13.70	18.40
	Static (latest/all)	Linear Regression	21.9 / 25.13	31.1 / 33.54
		Model Tree	18.20 / 14.56	25.8 / 19.02
		SVM	18.79 / 20.65	34.86 / 33.10
		Neural Network	17.32 / 20.96	21.81 / 24.97

4.3.4 Test with Data Generated from Non-Stationary Linear Mapping Function

24 data sets were generated randomly with assumed ranges of attributes and some trends reflecting real-world tablet PC leasing markets. Each data set has 200 instances and 8 different attributes shown in Table 6.4. The first part of Table 6.4 explains levels of each attribute which are decision variables explored in Section 4.2.2 The second part indicates an example of the generated data. The data set shows transactional history and the class variable is the price that customers paid. Since this is data from non-stationary linear mapping functions, mapping functions with some randomness vary over time.

The goal is to construct a series of predictive models for the $t = n+1$ data using $t = 1$ to $t = n$ data sets. n were increased by 1 from 11 to 23. For example, for an unseen $t = 12$ data set, static models were constructed using the latest data set $t = 11$ while CPTM mined a trend from $t = 1$ to $t = 11$ data sets and constructed a tree model from a predicted data set at $t' = 12$ with the calculated smoothing factor 0.5. Then, both models were evaluated with real $t = 12$ data set in terms of the MAE. This procedure continued up to $t = 24$ (total 13 times) for one time-ahead prediction and the results are shown in Figure 4.9.

In order to obtain a statistically valid conclusion between static models and CPTM, both parametric (F and T-test) and non-parametric (Mann-Whitney) tests were employed. With a significance level of alpha = 0.05, the accuracy of the CPTM model was significantly higher than that of static models with the generation of trend embedded data.

Table 4.9: Example of data set (decision variables and snapshot of data)

Display size (inch)	Weight (lbs)	Hard drive (GB)	CPU (technology)	Graphics card (technology)	Memory (GB)	Battery (hours)	Touchscreen (technology)	Price ($)
9	0.8	40	Core 2 duo	HD G	4	6	Touch D	p1
(Y_{11})	(Y_{21})	(X_{11})	(X_{21})	(X_{31})	(X_{41})	(X_{51})	(X_{61})	
10	1	80	Core 2 e	HD G 2000	6	12	Touch C	p2
(Y_{12})	(Y_{22})	(X_{12})	(X_{22})	(X_{32})	(X_{42})	(X_{52})	(X_{62})	
11	1.5	120	Core i3	HD G 2500	8	18	Touch B	p3
(Y_{13})	(Y_{23})	(X_{13})	(X_{23})	(X_{33})	(X_{43})	(X_{53})	(X_{63})	
12	2	250	Core 2 i5	HD G 3000	16	24	Touch A	⋮
(Y_{14})	(Y_{24})	(X_{14})	(X_{24})	(X_{34})	(X_{44})	(X_{54})	(X_{64})	
		320	Core 2 i7	HD G 4000	32			
		(X_{15})	(X_{25})	(X_{35})	(X_{45})			
		500	Core 2 i7 e					
		(X_{16})	(X_{26})					
10	1.3	40	Core 2 i7	HD G 2500	4	6	Touch D	950
10.5	0.8	80	Core 2 duo	HD G 2000	8	24	Touch B	910
12	0.9	320	Core 2 i5	HD G 4000	32	18	Touch C	1,200
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

4.3.5 Discussion

The CPTM algorithm showed good predictive performances in comparison to the four well-known static models in Table 4.8. From the cases of data sets generated from simple stationary linear and non-linear mapping functions, it is relatively clear to look at the effect of the geometric sampling and the time series prediction of attributes and class variables as shown in Figure 4.7, 4.8 and Table 4.7. The geometric sampling helped to reflect the trend of relation between attributes and class variables over time. The automatic time series forecasting also gave good approximations of future attribute and class values. In both cases, smoothing factors were close to zero, which makes sense in that stationary mapping functions were applied over time.

The real data in Section 4.3.3 had an interesting data structure. The values of attributes were fixed but the class variable continued to change, which is why there are only predictions for class variables in Table 4.7. It is important to realize that even though the forecast result was similar to the latest data, the geometric sampling improved the predictive performance alone. Empirical tests showed that without a precise prediction of attribute values, the CPTM algorithm worked well with the geometric sampling. Also, without any knowledge of customer preferences and their trend over time in the used product market, the selected smoothing factor can indicate that underlying relations between attributes and class variables were quite stationary in the interval of one month. This test with real data has great potential and can provide some directions from data collection to real application in different design domains.

The last case of data from non-stationary mapping functions represents a very complex data structure, and the question was whether CPTM worked well in this case. Due to the non-stationary nature of data, the prediction error

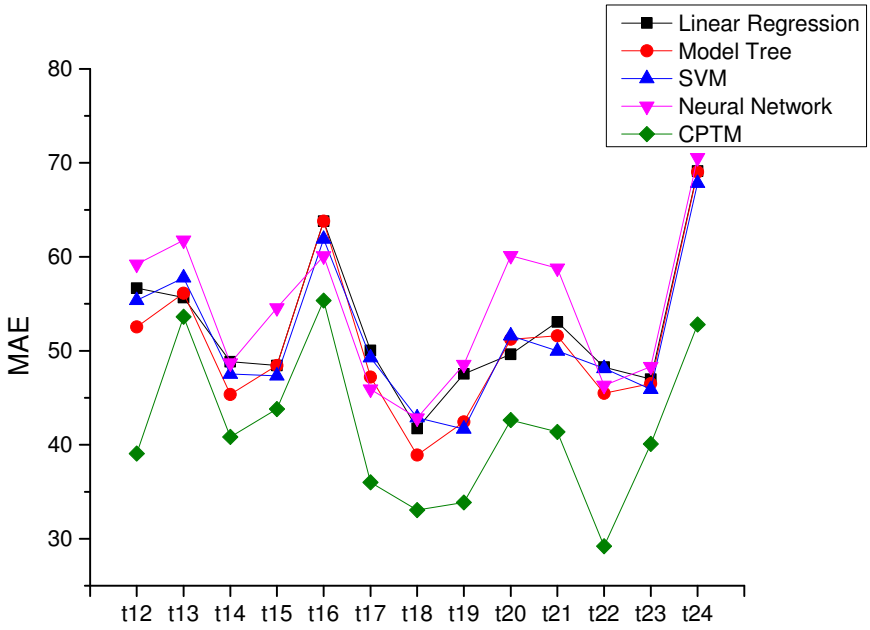


Figure 4.9: Comparison of the one time-ahead prediction accuracy between static and dynamic model (CPTM)

of CPTM was close to other static models, e.g., at $t = 13$ and $t = 14$, etc. in Figure 4.9. However, statistical results (Mann-Whitney test, $\alpha = 0.05$) showed that an overall performance of CPTM is better than other models with this data.

Among those four static models, the model tree was selected for the purpose of direct comparison since the CPTM algorithm also adopts the model tree for the prediction of class variables. From all the tested data cases, the predictive performance of CPTM outperformed that of the model tree, and this indicates that the generation of a trend embedded data set improved the accuracy.

While conducting these experiments, a total of five possible sources of variation on the result were observed: smoothing factor, model tree, time series prediction, selection of samples or random number generator, and size of samples. With the data sets from simple stationary linear and non-linear mapping functions, it is difficult to construct a good model due to the small number of instances. Random sampling can also have a great impact with the small sample size. However, the impact from the last two factors can be minimized with large-scale data. The model tree algorithm in CPTM is known to be fast or capable of dealing with large number of instances and attributes [131]. Empirical tests with a data set which is similar to the real buy-back price data (10 attributes) in Section 4.3.3 showed that the model tree took 1.34 seconds with 10^4 instances and 10.44 seconds with 10^5 instances running on an Intel Core i5 2.5 GHz Processor.

Moreover, the important observations are the facts that the model trees were pruned to avoid an over-fitting and

could generate performance information of the models by applying the 10-fold cross-validation technique. Also, continuous attributes and class variables were allowed in the models. These are aforementioned benefits of CPTM over the DPTM from generating the trend embedded future data.

4.4 Illustrative Example: Tablet PC Design

The overall methodology in Section 4.2 was applied to tablet PC design in the leasing market. The same data sets described in Section 4.3.4 were used. Weka and R in Section 4.3 also provided necessary tools for the model tree induction and the automatic time series forecasting.

4.4.1 Problem Setting

Tablet PCs are wireless, portable touch screen-operated computers. It is assumed that feasible candidate design attributes are defined in Table 6.4. It is expected that the start of leasing time is $t = 12$ and the company has accumulated data sets from $t = 1$ to $t = 11$. A manufacturer (and lessor) should manage multiple life cycles of its tablet PC by taking back leased products and re-leasing after processing for the next usage-life. The goal of this problem is to find the optimal tablet designs for multiple profit cycles while considering customer preferences, technological obsolescence, and reliability. Given inputs and assumptions are as follows:

Given inputs

· Historical transactional data as a set of attributes and price

· Generational difference information in Table 4.10

· Reliability information in Table 4.11

· Manufacturing and new parts cost in Table 4.12

· Reconditioning cost = $120

· Logistics cost: forward logistics= $5, reverse logistics= $5

Assumptions

· 2-year time frame (No disposal stage)

· Leasing period is fixed at six months

· After end-of-life processes, the reliability of a product will be back in a state as new

Table 4.10: Assumed information of generational difference

$t = 11$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	5	5	4	4	3	3
	4	4	3	3	2	2
	3	3	2	2	1	1
	2	2	1	1	0	0
	1	1	0	0		
	0	0				
$t = 12$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	6	5	4	5	3	3
	5	4	3	4	2	2
	4	3	2	3	1	1
	3	2	1	2	0	0
	2	1	0	1		
	1	0				
$t = 13$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	6	6	5	5	5	3
	5	5	4	4	4	2
	4	4	3	3	3	1
	3	3	2	2	2	0
	2	2	1	1		
	1	1				
$t = 14$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	8	7	6	6	5	5
	7	5	5	5	4	4
	6	4	4	4	3	3
	5	3	3	3	2	2
	4	2	2	2		
	3	1				
t=15	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	9	8	6	7	5	5
	8	7	5	6	2	4
	7	6	4	5	1	3
	6	4	3	4	0	2
	5	3	2	3		
	4	2				

Table 4.11: Assumed information of reliability

$t = 11$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	1	1	1	1	1	1
	1	1	1	1	1	1
	1	1	1	1	1	1
	1	1	1	1	1	1
	1	1	1	1		
	1	1				
$t = t + 1$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	0.95	0.95	0.96	0.98	0.99	0.99
	0.98	0.95	0.97	0.98	0.99	0.97
	0.99	0.98	0.98	0.99	0.99	0.97
	0.99	0.99	0.98	0.99	0.98	0.91
	0.99	0.99	0.93	0.96		
	0.95	0.93				

Table 4.12: Assumed information of cost for manufacturing and new parts ($)

Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
40	60	90	20	40	50
55	70	110	30	50	60
75	90	130	40	60	85
90	100	140	80	70	100
100	110	160	135		
120	125				

· Upgrade is not considered so that there are no compatibility issues during EOL processes

· Time for logistics and remanufacturing is negligible compared to that of the leasing period length

4.4.2 Applying CPTM

Since it was assumed that the tablet PC will have 6 months leasing time over the 2-year time frame, the number of life cycles is 4 and predictive models from $t = 12$ to $t = 15$ are needed by design. Static models were constructed using heuristics e.g., only the $t = 11$ data set or all historical data. For CPTM, it predicted 1 time, 2 time, 3 time, and 4 time-ahead data sets using $t = 1$ to $t = 11$ data sets selectively and built model trees from the predicted data sets. Table 4.13 presents the results. At $t = 12$ all split points of the 8 attributes were used to estimate SDRs of the class vales after splitting. Since the split point 2.5 of the attribute CPU maximized the standard deviation reduction of the class values, the CPU became the first node with branches of less than or equal to 2.5 and greater than 2.5. After all of the splitting, instances at the leaf nodes were used to build linear models. The resulted model tree was pruned so that it had only three linear models while the original tree had 169 linear models. By pruning the tree, the built model's performance was decreased based on the generated data or training data (e.g., 14.6 % more errors than the

unpruned tree) but the prediction accuracy of the model was improved with real $t = 12$ data (e.g., 1.3 % less errors than the unpruned tree) due to the generalization. The 10-fold cross-validation in Weka was used to get the performance information of the built model from the training data (i.e., the predicted data set at $t' = 12$) and the prediction accuracy was calculated from the real data (i.e., the real data set at $t = 12$). This shows that unpruned trees have a strong chance to be over-fitted. The comparison result between static models with the latest data set and CPTM is shown in Figure 4.10. Finally, these linear models will be used in the optimization model.

Table 4.13: CPTM results of illustration example

At $t = 12$, MT^{12} is defined as follows:
CPU $\leq$ 2.5 : LM1
CPU $>$ 2.5 :
| Hard Drive $\leq$ 2.5 : LM2
| Hard Drive $>$ 2.5 : LM3
LM num: 1
Class(Price) = -1.0472 * Weight + 0.6497 * Hard Drive - 8.8437 * CPU - 28.1698 * Graphics Card
- 17.0256 * Memory - 10.779 * Battery Life - 8.6369 * Touchscreen + 1014.3514
LM num: 2
Class(Price) = 7.8489 * Display size - 3.4677 * Weight - 12.11 * Hard Drive - 16.2778 * CPU
- 30.4275 * Graphics Card - 20.6762 * Memory - 3.8257 * Battery Life - 1.9893 * Touchscreen + 940.6956
LM num: 3
Class(Price) = 8.6802 * Display size - 3.4677 * Weight - 4.4711 * Hard Drive - 18.8712 * CPU
- 21.8863 * Graphics Card - 2.8183 * Memory - 32.5704 * Battery Life - 1.9893 * Touchscreen + 990.9373
At $t = 13$, MT^{13} is defined as follows:
Hard Drive $\leq$ 2.5 : LM1
Hard Drive $>$ 2.5 : LM2
LM num: 1
Class(Price) = 8.3753 * Display size - 33.9229 * Hard Drive - 20.2652 * CPU
- 37.4899 * Graphics Card - 11.9472 * Memory - 7.3387 * Battery Life - 8.3119 * Touchscreen + 993.0399
LM num: 2
Class(Price) = 0.7584 * Display size - 21.0345 * Weight - 6.1539 * Hard Drive - 18.1878 * CPU
- 8.9427 * Graphics Card - 7.4079 * Memory - 31.1933 * Battery Life - 0.505 * Touchscreen + 1103.8625
At $t = 14$, MT^{14} is defined as follows:
LM1
LM num: 1
Class(Price) = 16.3777 * Display size + 10.698 * Hard Drive - 17.8321 * CPU
- 20.8245 * Graphics Card - 14.2723 * Memory - 19.7954 * Battery Life + 853.3004
At t=15, MT^{15} is defined as follows:
LM1
LM num: 1
Class(Price) = 7.481 * Hard Drive - 27.5058 * CPU - 14.6723 * Graphics Card - 18.1991 * Memory
- 28.7345 * Battery Life - 6.7278 * Touchscreen + 1107.0882

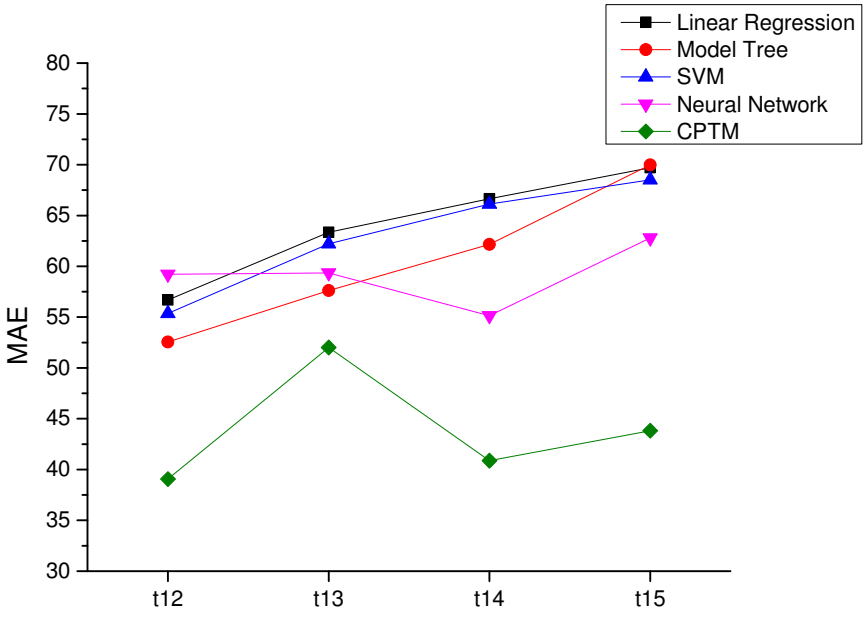


Figure 4.10: Comparison of 1, 2, 3 and 4 time-ahead prediction accuracy between static and dynamic model

4.4.3 Design for Multiple Profit Cycles

Table 4.14 shows the mathematical formulation of the illustration case derived from Section 4.2.2. The objective function consists of unit profits from four lease contracts and the interest rate, 3%, was assumed. Prices or market values that reflect the trend of customer preferences and technological obsolescence were formulated with the model tree functions depicted in Table 4.13.

Table 4.14: Mathematical formulation for illustration example

Objective Function
Maximize f=$(p^{12}-c^{12})$+ $\frac{1}{(1.03)^{0.5}}(p^{13}-c^{13})$+$\frac{1}{(1.03)}(p^{14}-c^{14})$+$\frac{1}{(1.03)^{1.5}}(p^{15}-c^{15})$
$p^{12}=MT^{12}(Y_{ij},X_{ij})$, $p^{13}=MT^{13}(Y_{ij},X_{ij})$, $p^{14}=MT^{14}(Y_{ij},X_{ij})$, $p^{15}=MT^{15}(Y_{ij},X_{ij})$
$c^{12}=\sum_j c_j^{manufacturing}X_{ij}+c^{forwardlogistics}$
$c^{13}=c^{inverselogistics}+c^{EOL(13)}+c^{forwardlogistics}$
$c^{14}=c^{inverselogistics}+c^{EOL(14)}+c^{forwardlogistics}$
$c^{15}=c^{inverselogistics}+c^{EOL(15)}+c^{forwardlogistics}$
Constraints
$h1:\sum_j Y_{ij}=1$
$h2:\sum_j X_{ij}=1$
$h3:c^{EOL(t)}=\sum_j c_j^{manufacturing}X_{ij}(1-\beta)+c^{reconditioning}\beta$
$h4:\beta=\prod_i(\sum_j \gamma_j X_{ij})$
$h5:Y_{ij},X_{ij}\in(0,1)$

4.4.4 Discussion

Similar to the previous section for the CPTM performance, Figure 4.10 indicates that the accuracy of CPTM outperformed that of those static models even with the multiple time-ahead predictions. The accuracy was measured by the mean absolute error which is the average deviation between predicted and observed class variable price. Based on Figure 4.10, the CPTM was adopted as the predictive model for this illustration.

The CPTM result in the Table 4.13 shows model trees constructed from the CPTM algorithm. At $t = 12$ and $t = 13$, multiple linear regression models were built and at $t = 14$ and $t = 15$, simple regression models were formulated to explain the class variable price. At $t = 12$, the model tree consists of three linear models: LM1, LM2, and LM3. When the attribute CPU has a generational difference less than 2.5, the first linear model, LM1, is selected. If the attribute CPU has a generational difference greater than 2.5 then the attribute hard drive will work as a splitting criterion. Again, if the attribute hard drive has a generational difference less than 2.5, then the second linear model, LM2, will be selected. Otherwise, the third linear model, LM3, will be used. In each linear model, 8 different design attributes in Table 6.4 with a constant term explain the class variable.

Excel solver with an evolutionary algorithm was used to solve the illustrated design problem. The selected designs are shown in Table 4.15, and the total life cycle unit profits are revealed in Table 4.16. The selected best design attributes are 12-inch in display size, 0.8-lbs in weight, 40-GB in hard drive, Core 2 i7 e in CPU, HD G 4000 in graphics card, 8-GB in memory, 24-hour in battery life, and touch C in Touch screen with the total life cycle unit profit of $1,562. There are other design results from linear regression models (i.e., static model) with the two heuristics in Table 4.15. First, the "only latest data set case" selected different CPU, memory, and touch screen attributes. It is interesting that the model generated much more profits at $t = 12$ but the design selected by CPTM brought more profits over the life cycle as shown in Table 4.16. Second, the "all data set case" selected different graphic card, memory, and touch screen attributes. This model generated more profits than the other heuristic but fewer profits than CPTM. The illustration concludes that the proposed framework can identify the optimal design that maximizes the total life cycle profit based on historical transactional data sets.

Table 4.15: Result of optimal tablet PC design

	Display size (inch)	Weight (lbs)	Hard drive (GB)	CPU (technology)	Graphics card (technology)	Memory (GB)	Battery (hours)	Touch screen (technology)
CPTM	12	0.8	40	Core 2 i7 e	HD G 4000	8	24	Touch C
Linear Regression (latest/all)	12 / 12	0.8 / 0.8	40 / 40	Core 2 duo / Core 2 i7 e	HD G 4000 / HD G 3000	32 / 32	24 / 24	Touch D / Touch A

From the result obtained from the artificially generated data, it can be argued that customers are very sensitive about the technological obsolescence for CPU, graphics card and battery attributes (refer to Table 6.4). Manufacturers

Table 4.16: Result of total life cycle unit profit

		$t = 12$	$t = 13$	$t = 14$	$t = 15$	Total life cycle
CPTM	Profit($)	419	430	386	327	1,562
	Price($)	949	994	951	875	3,769
	Cost($)	530	564	565	548	2,207
Linear Regression (latest/all)	Profit($)	477 / 392	390 / 418	346 / 376	232 / 343	1,445 / 1,529
	Price($)	962 / 972	919 / 1,000	875 / 958	745 / 909	3,501 / 3,839
	Cost($)	485 / 580	529 / 582	529 / 582	513 / 566	2,056 / 2,310

should use the latest cutting edge technology for these parts. At the same time, due to the popularity of cloud storage and external storage devices, the capacity of a hard drive seems to have become less important to customers. This suggests that manufacturers can place less priority on hard drive capacity. The proposed framework enabled this type of insight, which is not readily available under the previous trend mining approaches.

The illustration does not consider the option to upgrade for the initial design selection problem. However, an additional decision making process can determine the proper end-of-life options including upgrades [99]. Given the target time, manufacturers can decide whether the decrease of generational difference (i.e., upgrade) is better than reconditioning for each component. The life cycle management plan can then be set up including upgrades.

4.5 Conclusion

In this chapter, a new predictive trend mining algorithm, CPTM, is developed in the context of product and design analytics. Unlike traditional, static data mining algorithms, CPTM does not assume stationarity, and dynamically extracts valuable knowledge of customers over time. By generating trend embedded future data, the CPTM algorithm not only shows higher prediction accuracy in comparison with static models, but also provides essential properties that could not be achieved with the previous trend mining algorithms: dynamic selection of historical data, avoidance of over-fitting problem, identification of performance information of constructed model, and allowance of a numeric prediction. Various generated data sets and the real data set were used to test the performance of CPTM and those benefits were verified. Also the optimization model for multiple life cycles is formulated as a binary integer programming model and combined with the CPTM result. Using the proposed framework, design engineers can select the optimal design for the target product that can generate multiple profit cycles. The illustration example of tablet PC design showed that the optimization model with CPTM can reveal hidden profit cycles successfully.

It will be interesting to observe the impact of the prediction interval in the CPTM algorithm even though there are multiple sources of variation as discussed in Section 4.3.5. Different optimal design solutions can be obtained based on the interval. The optimization model in the illustration example was simplified in order to show the application

of CPTM. Additionally, compatibilities among different parts, different product life cycles, and product families can be considered for more interesting and realistic problems. Finally, instead of having a set of attributes as a priori, capturing of emerging attributes and management of dynamic attribute sets would be possible tasks in the future.

The next chapter will discuss optimal design in the area of product family design. Since the product family design of products that can be highly shared by many other products is very data-intensive, predictive design analytics can provide a new insight for this design problem. Prediction intervals will be incorporated in the process of decision making, and customer preferences to affect the determination of product architectures will be identified from data. A large volume of data will be utilized in an illustrative example to test the algorithm can handle the large-scale data.

Chapter 5

Product Family Architecture Design with Predictive, Data-Driven Product Family Design Method

This chapter[1] addresses the challenge of determining optimal product family architectures with large-scale customer preference data. The proposed model, predictive data-driven product family design (PDPFD), expands clustering based data-driven approaches to incorporate a market-driven approach. The market-driven approach provides a profit model in the near future to determine the optimal position and number of product architectures among product architecture candidates generated by the k-means clustering algorithm. Unlike discrete choice analysis models which were used in previous market-driven approaches, a market value prediction method is proposed as a dynamic model which can capture and reflect the trend of customer preferences. Prediction intervals provide market uncertainties of the dynamic profit model for product architecture design. A universal electric motors design example is used to demonstrate the implementation of the proposed framework with large-scale data. The comparative study shows that the PDPFD algorithm can generate more profit than pure clustering based data-driven models, which shows the necessity of combining data-driven and market-driven approaches

5.1 Introduction

Today's highly competitive market situation and enormous data generation environment mean companies and design engineers have to consider a wide variety of customer preferences and requirements. Massive-scale customer preference data is available from various data sources such as company databases, social networks, clickstreams, etc. In order to accommodate the diversity of customer preferences, designing a family of products becomes a prevailing strategy across many industries [135, 45, 41].

The main question considered in this study is how to determine the optimal product family architectures with large-scale customer preference data. Clustering based data-driven methodologies [64, 65] were presented to identify central points of clusters (market segments) in the customer preference space (performance requirements). The central points are *ideal points* in market segments [65] and are also product family architecture candidates. Tucker et al. [64] proposed that a clustering technique can enable design engineers to identify the optimal number of product architec-

[1] Presented in [132, 133] and submitted to [134].

tures from large-scale customer preference data. This study expands these clustering based data-driven approaches [64, 65] to incorporate a market-driven approach [42, 61] with massive-scale customer preference data. The market-driven approach provides a profit model as an objective function to determine optimal product architectures. Unlike the previous market-driven approaches [42, 61], this study does not assume that 1) market segments are given, and 2) customer preferences are static (i.e., no change over time).

The products of interest are products or parts that can be highly shared by many other products, including universal motors in power tools and home appliances, engines in on and off-road vehicles, batteries in electronics, etc. These products should satisfy a wide variety of different customers' requirements. The product family design scenario that this study focuses on is presented as follows. A company wants to analyze historical large-scale transactional data in order to support its product family architecture decision for new orders. Figure 5.1 shows a data-driven approach in the two-dimensional (requirement 1 and 2) customer requirement (circles) space. The objective is to determine the position and number of product architectures (e.g., one rectangle and three triangles) in order to satisfy customers' requirements. Pure data-driven approaches might generate geometrically meaningful results. For example, the product architecture in the middle (rectangle) can be the optimal solution based on the selected information criterion (model fitting function with penalization of complex models) but it might end up with an inferior solution from the perspective of markets. With the guidance of market-driven approaches, data-driven approaches can produce a meaningful result for decision makers. Once architectures are determined then clusters can be interpreted as market segments (dotted lines).

Figure 5.2 shows a market-driven approach, which evaluates product architecture candidate sets (one rectangle and three triangles in Figure 5.1) in terms of profit. With estimated revenue and production cost, the profit and its uncertainty (dotted line) can be estimated. Note that the X-axis represents the number of product architectures and the Y-axis represents the monetary value. When the number of product architectures is increased, the fixed costs will be increased with more product variants. However, since more customers' product requirements can be satisfied, revenue can be increased too. Figure 5.1 and 5.2 together show the necessity of a market-driven approach in the clustering based data-driven approaches.

Predictive, data-driven product family design (PDPFD) proposed in this study aims to merge data-driven and market-driven approaches based on the predictive design analytics paradigm. The predictive design analytics paradigm [81, 115] enables design engineers to extract knowledge from large-scale, multidimensional, unstructured, volatile data, and transform that knowledge and trend into design decision making. The PDPFD framework introduces predictive profit modeling in a clustering based data-driven model so that it can support complex product family architecture decisions. In order to capture trends in profit, market value prediction with regression coefficients is used together with time series analysis. The proposed method is demonstrated using a universal motor design problem [3] with massive-

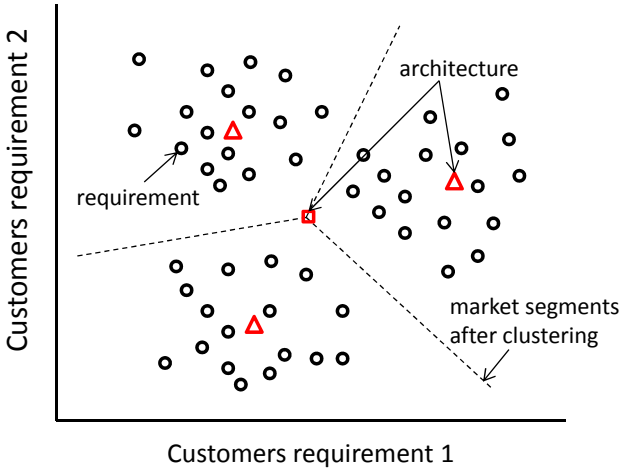


Figure 5.1: Overview of data-driven approach

scale customer preference data. Finally, a previous data-driven method [64] is compared to the PDPFD method in order to show the benefits of the proposed method.

The rest of the study is organized as follows: The proposed approach, PDPFD, is presented in Section 5.2 followed by a case study in Section 5.3. The benefits and limitations of the proposed approach along with future work are discussed in Section 5.4.

5.2 Methodology

5.2.1 Overview

Figure 5.3 outlines the framework of PDPFD. There are two stages: individual product design stage and product family design stage. The individual product design stage involves the enterprise level and engineering level [59, 60, 136]. The enterprise level represents managerial level decision making for maximizing the expected profit with respect to the number and specifications of architectures as targets. The engineering level represents physical design decision making with respect to engineering level design variables (e.g., thickness and length of parts). The objective function consists of local objective functions (e.g., minimizing product's weight) and the deviation term for target matching (e.g., satisfying performance requirements). If the enterprise level target is infeasible, then a new target should be explored. Once the individual product design stage decisions are made, the next step is to deterime product family design. Based on the determined product variants, a decision making process for scale-based product family design is explored. Scaling variables (i.e., the reduced design variables) of the architectures can be stretched or shrunk to satisfy the same objective function in engineering level while common parameters remain constant. The common parameters

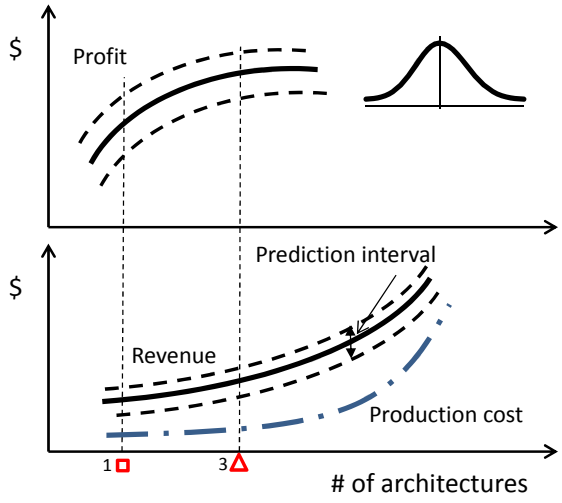


Figure 5.2: Overview of market-driven approach

constitute the product platform.

Three important tools for the PDPFD framework are a market value prediction model with exponential smoothing for market considerations (Section 5.2.3), k-means clustering for product family architecture candidates (Section 5.2.4), and expectation maximization clustering for multiple-platform design (Section 5.2.5). The first tool will capture a trend of customer preferences and uncertainties, the second tool will find the optimal number of architectures to minimize deviations between customer requirements and performance of architectures, and the last tool will figure out the possibility of multiple platforms.

5.2.2 Data Structure and Assumptions

The main question in a data-driven model is how to represent data. Figure 5.4 shows the basic data structure. The index t represents discrete time and data at $t = n$ indicates the current data. In the historical data set from $t = 1$ to $t = n$, transactional information is available, which is the set of data on product requirements (e.g., torque and efficiency), chosen product architectures (e.g., a1, a2, etc), and the discounted price that customers paid based on their utility for the chosen product architecture. Note that discounts can be applied if the product requirements cannot be matched. The goal is how to determine the position and number of product architectures at h time-ahead (i.e., at $t = n + h$). Furthermore, the trend in customer preferences in historical data is captured and reflected in a profit function.

The transaction tables in Figure 5.4 also show the generation of the deviation between what customers want and what products provide. By generating the deviation columns from product requirements and product architectures, the impact of increasing or decreasing product architectures can be investigated in terms of discounts.

The availability and quality of data are critical in data-driven models. The data set utilized in this study is transac-

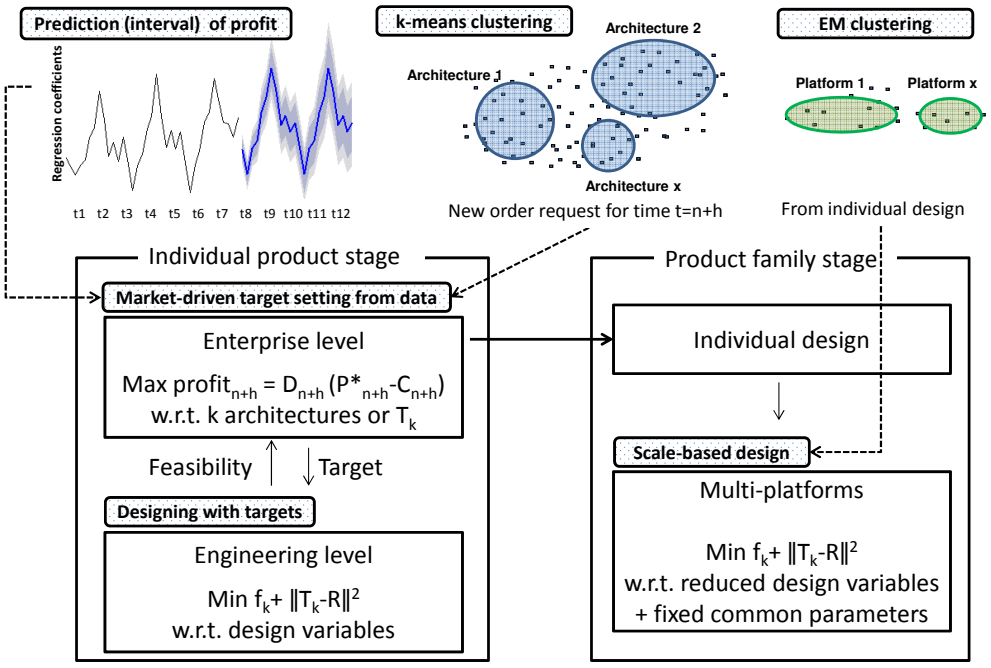
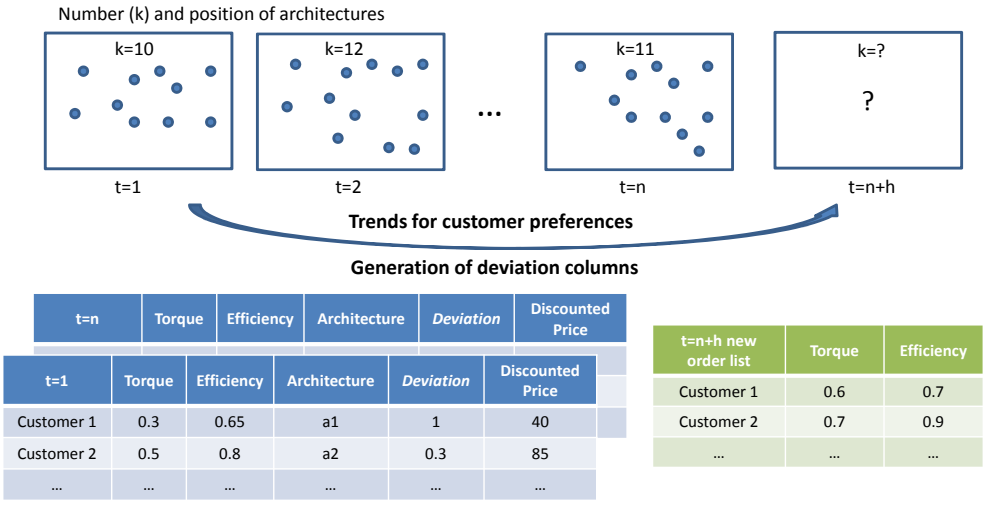


Figure 5.3: Overall framework of PDPFD

tional information, which can be found in company databases though it is usually classified as confidential. Instead of directly analyzing real data sets, randomly generated data sets will be used to test the proposed model. Since the quality of data-driven models can be hugely affected by the quality of data, great efforts should be made for the preparation of input data sets. To improve the quality of data, data cleaning methods were adopted such as removing abnormality values and handling missing values [119].

The basic assumptions in the framework of PDPFD are depicted in Figure 5.5. The circles represent customers' requirements in terms of performance of products, and the rectangle shows the centroid of the cluster or the architecture. In the extreme case, seven product architectures can be developed to satisfy all customers, which is the ideal case of mass customization. Or, only one product architecture (the current figure) can serve as a single medium to embrace all the requirements if the customers can ignore the differences. It is assumed that customers will buy the product that is closer to their requirements in terms of the Euclidean distance. Basically, the performance of the product will determine price and cost functions. For example, key performances of notebook computers (e.g., memory, processors, screen size, etc.) determine notebook computer price and cost. In addition, the deviation or distance between a product architecture and customer requirements will affect a purchase in terms of the discounted price, and the increasing the number of architectures will increase the fixed costs.

Under the aforementioned assumptions, the result of the PDPFD framework can be used in a product design



t=n	Torque	Efficiency	Architecture	*Deviation*	Discounted Price
t=1	Torque	Efficiency	Architecture	*Deviation*	Discounted Price
Customer 1	0.3	0.65	a1	1	40
Customer 2	0.5	0.8	a2	0.3	85
...	...	...	...	...	...

t=n+h new order list	Torque	Efficiency
Customer 1	0.6	0.7
Customer 2	0.7	0.9
...	...	...

Figure 5.4: Example of data structure

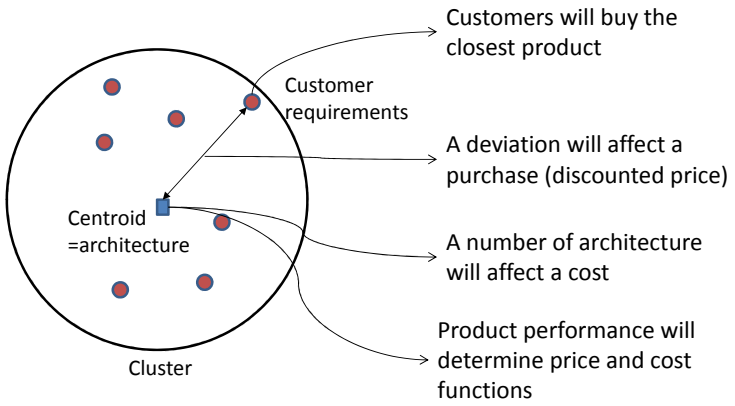


Figure 5.5: Basic assumptions of PDPFD

decision support system. No competing product is considered so that the impact of product brand is not investigated in this study. Also, product performance in the customer requirement space is limited to continuous variables. The proposed model attempts to model the trend of customer preferences in the market and use the trend and prediction intervals for the product design decision support system. Since the predicted model is designed to be used for a short forecasting horizon (e.g., one-step-ahead short prediction such as three months and six months later), the evolution of a product family and technology shifts are not considered.

5.2.3 Market Value Prediction for a Profit Model

In order to build a predictive profit model (Section 5.2.4), market value prediction is the key component and this section will provide the method with prediction intervals (i.e., lower and upper bounds). Most of all, significant factors among identified design requirements for prices and costs should be identified. Subject matter experts are helpful to manage the list of candidate factors, and stepwise regression procedures can be applied to find the significant factors in a stepwise manner.

Market value prediction with regression coefficients

Prediction of product prices with regression coefficients was proposed by Rutherford and Wilhelm[137] for a notebook computer (hereinafter RW model). Recently, this model was revalidated with a more mature notebook market [138]. Though the RW model was validated with a notebook computer, it was also used to relate demand, price, and the features that comprise a general product [139, 140] and suggested as a possible prediction method of product design [141]. The RW model consists of two phases. Phase 1 fits a linear regression model to each time series. Phase 2 uses linear trend analysis of regression coefficients to capture a trend over time. Then, future market values of target products can be predicted with given features. From publicly available data (notebook price data), the model predicted the rate of price erosion of a notebook computer up to seven months ahead within 10% error. The RW model is used for the base case of price prediction.

The main difference between the RW model and the predictive model in PDPFD (hereinafter PDPFD model) is that the PDPFD model uses exponential smoothing models at Phase 2, which is more flexible (e.g., linear trend model can be considered one of exponential smoothing models) and provides prediction intervals for prediction uncertainty. The general form of the regression model in this study is given in Equation (5.1):

$$P_t = \beta_{0t} + \sum_{i \in A} \beta_{it} a_{it} + \theta_t, \text{ for } t = 1, \cdots, n \tag{5.1}$$

where P_t is the price or market value of a product at discrete time t, β_{0t} is the intercept, i is the index for levels or alternatives of product features, A is the set of product features, β_{it} is the regression coefficients of factor i, a_{it} is the measurement of factor i, and θ_t is the random error. Note that the price is determined by product features but the discounted price considers one more factor, diviation in Section 3.4. It does not need to be linear but homogeneous forms of regression models are required over time (i.e., linear, squared, cubic, etc.) to apply the PDPFD model. Linear regression is usually adopted as a general model with the following assumptions: linear relationship between factors and response, independent factors and random errors, and random error with constant variance.

The next step is to trace the trend of β_{it}, which is considered as customer preferences over time. Exponential

smoothing based on innovations state space models [96] is proposed to model the time series. Equation (5.2) and (5.3) show generalized state space equations for β_{it} [96]:

$$\beta_{it} = w(x_{i(t-1)}) + r(x_{i(t-1)})\varepsilon_{it} \tag{5.2}$$

$$x_{it} = f(x_{i(t-1)}) + g(x_{i(t-1)})\varepsilon_{it} \tag{5.3}$$

where β_{it} is the observed value at time t, x_{it} is the state vector which contains unobserved components such as the level, trend, and seasonality of a time series, $w()$ and $r()$ are scalar functions, $f()$ and $g()$ are the vector functions, and ε_{it} is the white noise process with variance σ^2. The white noise process has zero mean, constant and finite variance, and uncorrelated values. For a succinct notation, index $i \in A \cup \{0\}$ is used in Equation (5.2) and (6.15).

By combining Equations (5.1), (5.2) and (5.3), the following state space based price equations are formulated:

$$P_t = [w(x_{0(t-1)}) + r(x_{0(t-1)})\varepsilon_{0t}] + \sum_{i \in A} [w(x_{i(t-1)}) + r(x_{i(t-1)})\varepsilon_{it}]a_{it} + \theta_t \tag{5.4}$$

$$x_{it} = f(x_{i(t-1)}) + g(x_{i(t-1)})\varepsilon_{it} \tag{5.5}$$

Finally, estimation of the price at h time-ahead is formulated as follows:

$$\hat{P}_{t+h} = \hat{\beta}_{0(t+h|t)} + \sum_{i \in A} \hat{\beta}_{i(t+h|t)} a_{i(t+h)} \tag{5.6}$$

where $\hat{\beta}_{t+h|t}$ represents the forecast of $\hat{\beta}_{t+h}$ based on all the data up to time t.

There are a total of 30 exponential smoothing models classified based on trend, seasonality, and error in additive, multiplicative or mixed ways. Hyndman et al. [96] provided details of the classifications. The automatic forecasting method [127] is adopted to determine all the necessary parameters and the best model. The first step is to apply all the 30 exponential smoothing models, and estimate initial states and parameters using maximum likelihood estimation based on the innovations representation of the probability density function (refer to Equation (5.8)). The next step is to choose the best model according to an information criterion: Akaike's information criterion (AIC), corrected Akaike's information criterion (AICc) or Bayesian information criterion (BIC).

Prediction interval of market value

In the previous section, point forecasting of the time series β_{it} was discussed, which provides an average market value of products. In order to consider the uncertainty in market trends, prediction intervals in time series prediction are

used as well.

Three sources of uncertainty were identified in forecasting a future value [96]: 1. selected model, 2. estimated parameters and initial states, 3. future innovations: $\varepsilon_{i(n+1)}, \cdots, \varepsilon_{i(n+h)}$. If it is assumed that the uncertainties from the first and second sources can be minimized by applying the automatic forecasting method in Section 5.2.3, the uncertainty in the future innovations is the only source that needs to be considered for prediction intervals.

If the initial state value x_{i0} is known, the innovation ε_{it} is a one-step-ahead prediction error. The conditional expectation [96], which is also the one-step-ahead point forecast $\hat{\beta}_{it|(t-1)}$ is given by:

$$E(\beta_{it}|\beta_{i(t-1)}, \cdots, \beta_{i1}, \beta_{i0}) = E(\beta_{it}|x_{i(t-1)}) = \hat{\beta}_{it|(t-1)} = w(x_{i(t-1)}) \tag{5.7}$$

The probability density function [96] for β_i is also given as a function of innovations ε_{it} in Equation (5.8):

$$P(\beta_i|x_{i0}) = \prod_{t=1}^{n} P(\beta_{it}|x_{i(t-1)}) = \prod_{t=1}^{n} P(\varepsilon_{it})/r(x_{i(t-1)}) \tag{5.8}$$

Then, the recursive relationships can be summarized as follows:

$$\hat{\beta}_{it|(t-1)} = w(x_{i(t-1)}) \tag{5.9}$$

$$\varepsilon_{it} = (\beta_{it} - \hat{\beta}_{it|(t-1)})/r(x_{i(t-1)}) \tag{5.10}$$

$$x_{it} = f(x_{i(t-1)}) + g(x_{i(t-1)})\varepsilon_{it} \tag{5.11}$$

Therefore, h time-ahead prediction of β_{it} requires only $\varepsilon_{i(n+1)}, \cdots, \varepsilon_{i(n+h)}$.

In order to obtain prediction distributions, a simulation approach [96] is adopted, which is simple and can cover all the 30 exponential smoothing models. The simulation approach simulates sample paths or observations β_{it} with initial states x_{it} from the chosen model. The remaining unknown values are future innovations ε_{it}, and they can be obtained from a random number generator with an appropriate distribution. An approximate $100(1-\alpha)\%$ prediction interval for forecast horizon h is given by the $\frac{\alpha}{2}$ and $1-\frac{\alpha}{2}$ quantiles of $\beta_{i(t+h)|t}$:

$$\hat{P}_{t+h}^{\frac{\alpha}{2}} = \hat{\beta}_{0(t+h|t)}^{\frac{\alpha}{2}} + \sum_{i \in A} \hat{\beta}_{i(t+h|t)}^{\frac{\alpha}{2}} a_{i(t+h)} \tag{5.12}$$

$$\hat{P}_{t+h}^{(1-\frac{\alpha}{2})} = \hat{\beta}_{0(t+h|t)}^{(1-\frac{\alpha}{2})} + \sum_{i \in A} \hat{\beta}_{i(t+h|t)}^{(1-\frac{\alpha}{2})} a_{i(t+h)} \tag{5.13}$$

For example, 90% of the prediction interval of a market value is given by $\hat{P}_{t+h}^{0.05}$ and $\hat{P}_{t+h}^{0.95}$. The prediction interval

should be interpreted as the average prediction success instead of any single case. In other words, 90% of the time, the real market value will fall within the bounds of intervals.

Performance test for predictive model in PDPFD

In this section, the prediction capabilities of the PDPFD model and the RW model in Section 5.2.3 are compared. The hypotheses are 1) the PDPFD model can provide a similar level of predictive accuracy to the RW model when data has a simple trend (trend of regression coefficients) and 2) the PDPFD model can predict future values more accurately than the RW model when data has complex patterns (e.g., trend and cycle of regression coefficients).

Data sets with a simple trend and complex patterns were generated randomly with the description of the generation procedures. Each data set contains three factors and one class variable (response or dependent variable) with 100 instances. The goal is to predict one-step-ahead class values using previous data sets. There were a total of 30 data sets from t=1 to t=30 and the prediction results were collected from t=11 to t=30 (i.e., 20 time periods).

As a performance measure, mean absolute error (MAE) was selected as given by Equation (5.14):

$$\text{Mean Absolute Error} = \frac{|b_1 - d_1| + \cdots + |b_m - d_m|}{m} \tag{5.14}$$

where $b_1, b_2, \cdots, b_m$ are the predicted class values and $d_1, d_2, \cdots, d_m$ are the actual class values.

Data with trend

For the first hypothesis, the following data generation procedure was applied: 1) the value of each factor was randomly chosen from 1 and 5 for each of the 30 data sets, 2) the base regression coefficients (i.e., t=1) for three factors were randomly chosen between 30 and 40, 3) one of possible trends (increasing 1.5 or decreasing 1.5) was randomly selected and applied to each coefficient from t=2 to t=30, 4) the class values were generated based on the values of the factors and the regression coefficients with some additional randomness, 5) the regression analysis was applied to the generated data sets, 6) the identified values of regression coefficients were used for predictive modeling. Due to the randomness in step 4, the trend of regression coefficients are not exactly 1.5.

The result of 20 MAEs (each MAE represents the average of absolute errors for 100 instances) showed that the prediction accuracies of the PDPFD method and the RW model were almost identical (Mann-Whitney test, $\alpha = 0.05$, p-value=0.98). Both models predicted one-step-ahead values with less than 1% error.

Data with trend and cycle

For the second hypothesis, the following data generation procedure was applied: 1) the value of each factor was

randomly chosen from 1 and 5 for each of the 30 data sets, 2) the base regression coefficients for three factors were randomly chosen with cyclical patterns (e.g., t=1 between 30 and 40, t=2 between 40 and 50, t=3 between 50 and 60, t=4 between 60 and 70), 3) one of possible trends (increasing 1.5 or decreasing 1.5) was randomly selected and applied to the regression coefficients of each cycle from t=5 to t=30, 4) the class values were generated based on the values of the factors and the regression coefficients with some additional randomness, 5) the regression analysis was applied to the generated data sets, 6) the identified values of regression coefficients were used for predictive modeling. As a result of this procedure, similar patterns were repeated for every four-time steps (i.e., cycles).

Table 5.1 shows the comparison result from both models. Since the RW model depends only on the trend line for the prediction, when data has complex patterns, the PDPFD model provides a higher prediction accuracy (Mann-Whitney test, $\alpha = 0.05$, p-value=0).

Table 5.1: Comparison between RW and PDPFD model over 30 data sets (MAE)

	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20
RW	25.5	142.6	213.5	49.5	25.5	149.7	201.9	49.4	26.2	162.3
PDPFD	2.7	3.4	2.8	2.9	3.1	2.6	2.8	3.0	3.0	3.4
	t21	t22	t23	t24	t25	t26	t27	t28	t29	t30
RW	183.9	46.1	27.4	156.8	180.9	44.7	28.7	161.4	177	44.2
PDPFD	2.9	3.0	3.2	3.1	3.0	3.0	2.8	2.6	2.9	2.6

* MAE: mean absolute error

The strength of the PDPFD model comes from the fact that both linear and non-linear forms of formulations can be used, and the trend of coefficients can be captured dynamically in an automatic way. Moreover, the PDPFD model can provide prediction intervals (e.g., forecast value is 60.3 with 80% prediction interval of 59.8 and 60.8), which can show the uncertainty of market trend (customer preferences) in Section 5.2.3. These are characteristics of the predictive model in PDPFD, which are not present in the RW model.

Now, a general model of predicted market values and its interval is formulated and tested. In the next section, the model will be combined with a profit model.

5.2.4 Individual Product Design Stage

In the individual product product stage, there are two levels: enterprise level and engineering level [59, 60, 136]. As shown in Figure 5.3, the market-driven target setting from large-scale customer preference data is implemented at the enterprise level, and engineering design with the target is realized at the engineering level.

Enterprise level

At the enterprise level, the objective is to maximize the expected profit while satisfying other constraints:

Maximize

$$\Pi_{n+h}(T_k) = D_{n+h}(P^*_{n+h} - C_{n+h}) \tag{5.15}$$

Subject to:

$$g(T_k) \leq 0, h(T_k) = 0 \tag{5.16}$$

where Π_{n+h} is the economic profit at time $n+h$ (h time-ahead); T_k is the set of target values (i.e., product architectures with k number); D_{n+h} is the demand or number of orders; P^*_{n+h} is the discounted price or sale price; C_{n+h} is the cost; $g()$ are inequality constraints (e.g., range of k or minimum profit); $h()$ are equality constraints (e.g., exact number of k).

Equation (5.17) and (5.18) show the general models for the price and cost based on the assumptions in Section 5.2.2:

$$P^*_{n+h} = f(T_k, d) \tag{5.17}$$

$$C_{n+h} = f(T_k, k) \tag{5.18}$$

where $f()$ is a scalar function; d is the deviation in Equation (5.20), which represents the impact of deviations between customers' requirements and product architectures; k is the number of architectures, which represents fixed costs to increase the number of architectures. In order to apply regression analysis, it is assumed that historical data has $k \geq 2$. The data for the cost model at $t = n+h$ is assumed to be available to manufacturers but the price model at $t = n+h$ should be predicted as discussed in Section 5.2.3. If cost related data at $t = n+h$ is not available, the same technique used in the price model should be applied.

To solve this problem with large-scale data, a two-step approach is proposed. The proposed process starts from identifying maximum k. Then, find each $T_2, \cdots, T_k$ that minimizes deviations from customer requirements. Next, among $T_2, \cdots, T_k$, determine the best one by considering profit prediction along with its prediction intervals at the target time. Note that since this is product family design, more than two product variants (T_2) will be realized.

Step 1: set maximum k or number of architectures, and calculate a deviation for all k centroids by applying k-means clustering

Step 2: calculate profits for all k architectures with prediction intervals, and set the target T_k that generates maximum profit

The determination of maximum k in this algorithm depends on designers. In general, it is almost impossible for designers to decide the number from large-scale data. However, the maximum number of architectures (k) can be estimated not purely by data but jointly by manufacturer's capability and managerial decisions (e.g., the number of production lines allow only a certain number of product variants). If the maximum number k cannot be estimated, k should be increased enough to the point where no more improvement is possible in the case of a concave profit function. Tucker et al. [64] used the X-means clustering algorithm to automatically select the optimal k for product family architecture design but the maximum k should be provided by designers.

The k-means clustering algorithm [142, 119] is used since it is simple and effective. The Euclidean assumption in Figure 5.5 works well with the k-means algorithm. The clustering algorithm partitions a given data set into a fixed number of clusters k. It aims at minimizing the objective function, which is within cluster sum of squared errors (SSE) as shown in Equation (5.19):

$$f = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 \tag{5.19}$$

where $x = (x_1, x_2, \cdots, x_n)$ is a set of customer requirements; $C_i = (C_1, C_2, \cdots, C_k)$ is a set of clusters; c_i is the centroid of cluster C_i (which is the arithmetic mean of points in C_i). The deviation d is defined in Equation (5.20):

$$d = \frac{\sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2}{n} \tag{5.20}$$

The iterative process of the k-means algorithm starts by specifying the number of clusters (k). Then, k points are chosen randomly as cluster centers (c_i) and all instances (x) are assigned to the closest cluster centers in accordance to the Euclidean distance. After the assignment, new cluster centers are recalculated as means. This process is repeated until the same instances are assigned to the same clusters.

The k-means clustering algorithm has some disadvantages as follows. First, it is necessary to specify the number of cluster k by designers. It was discussed above how to constrain the k for product family architecture design. Second, its performance can be significantly diminished with high dimensional data. New k-means clustering algorithm with high dimensional data was proposed by Sun et al. [143] and various dimensionality reduction techniques were discussed in the literature such as principle components analysis [119], kernel trick [119], data compression [65], feature selection [119, 64], etc. If data is really high dimensional (e.g., DNA, tweets, etc.) special clustering techniques should be applied [144]. Third, the algorithm converges to local minima. Initial starting points can affect the result and repeating the algorithm with different starting points is required. Note that these disadvantages are common in any clustering algorithm.

Engineering level

The engineering level problems can be stated as follows: find a design solution that minimizes the deviations between design targets from Section 5.2.4 and actual responses while satisfying design constraints:

Minimize

$$f_k + \|T_k - R\|_2^2 \tag{5.21}$$

Subject to:

$$g(T_k) \leq 0, h(T_k) = 0 \tag{5.22}$$

where f_k is the local product design objective function(s) (e.g., minimize weights); T_k is the target vector cascaded down from the enterprise level; the R is the response vector obtained from the analysis model $r(x)$ (e.g., engineering level analytical models to calculate the response of the targets).

5.2.5 Product Family Design Stage

The goal of the product family design stage is to find clusters of values under each common parameter for exploring the possibility of multiple platforms while maintaining the performances of products. The clustering is based on similarity without the prior knowledge of cluster numbers. There are a few clustering techniques to allow this task: expectation maximization (EM) [145, 119, 146] and X-means clustering [147]. Both of them are extended versions of the k-means clustering method, which is used in the individual product design stage. Based on empirical test results for the product family design stage, the EM algorithm is used in this stage.

The EM clustering algorithm is a generalization of maximum likelihood estimation when the given data set is incomplete or there are unobserved latent variables. The goal is to estimate parameter $\hat{\theta}$ that maximizes the log-likelihood $\log P(x, z; \theta)$, where x is the observed variable and z is the latent variable. The EM iteration alternates between the expectation (E) step, which calculates a probability distribution over possible completions of missing data with the initial guess of parameters, and the maximization (M) step, which re-estimates the parameters using these completions. Do and Batzoglou [146] provided a simple coin-flipping example of the EM algorithm.

In the clustering task, the unobserved latent variables are the assignments of observed values to clusters, and the parameters are the means and covariance matrices of the selected distributions representing each cluster. Therefore, the E-step calculates the cluster probabilities with the guessed parameters. The M-step calculates the parameters (i.e., cluster means and covariances) by maximizing the likelihood of the distributions.

Based on the result of the EM clustering, multiple values are allowed for common parameters. Whether one constant (i.e., single platform) or multiple constant values (i.e., multiple platforms) are used for common parameters

depends on designers. Finally, the engineering level optimization problem should be re-solved with respect to reduced design variables (i.e., scaling variables) with fixed common parameters.

5.3 Illustrative Example: Universal Motor Family Design

5.3.1 Background and Data Generation

The design of a universal motor family [3] is used to demonstrate the effectiveness of the proposed model and provide a comparison of the proposed model and a pure clustering based data-driven model [64]. Universal electric motors are the most common components in power tools such as electric saws, drills, drivers, etc. and in household appliances such as blenders, vacuum cleaners, washing machines, etc. Figure 5.6 shows the schematic of a universal motor. There are eight design variable as inputs in Table 5.2. A mathematical model provided by Simpson et al. [3] returns four performance outputs: power (P), torque (T), mass (M), and efficiency (η) of motors as a function these eight design variables. The objective of this case study is designing a family of universal electric motors that maximizes the expected profit for the next market trend (customer preferences) based on accumulated large-scale data.

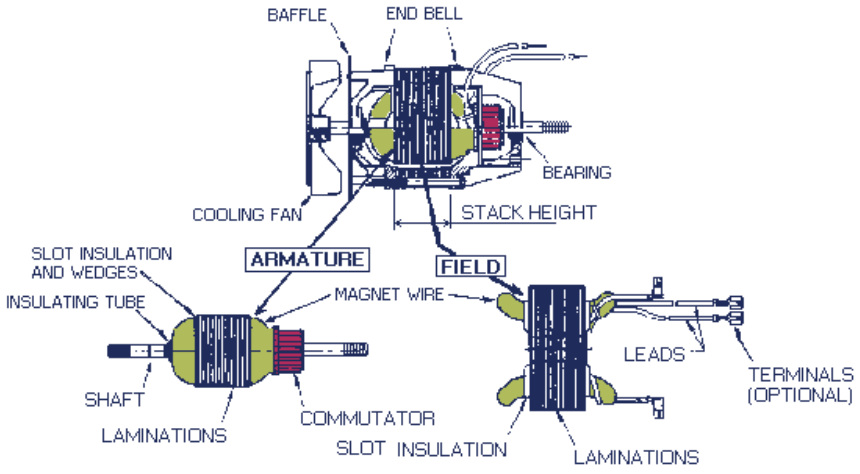


Figure 5.6: Universal motor schematic (source: [3])

Three large-scale data sets (*data set 1*, *data set 2* and *data set 3*) were generated using the generation procedure in Section 5.2.3 (Data with trend) with manually generated new orders. Figure 5.7 shows the new orders in *data set 1* and *data set 2*, which needs to be clustered. Due to the security issues with real data, the simulated data sets were used to test the proposed model. Each data set contains twelve historical (six-month interval) transactional data, one new order data, and one cost related data. Each data has one million instances (i.e., a total of 14 million instances for each data set). The embedded artificial trends in *data set 1* is shown in Table 5.3. For example, the coefficients of efficiency

Table 5.2: Design variables and ranges of universal motors

Variable	Definition	Range
N_c	Number of wire turns on the motor armature	$100 \leq N_c \leq 1500$ turns
N_s	Number of wire turns on each field pole	$1 \leq N_s \leq 500$ turns
A_{wa}	Cross-sectional area of the armature wire	$0.01 \leq A_{wa} \leq 1\ mm^2$
A_{wf}	Cross-sectional area of the field wire	$0.01 \leq A_{wf} \leq 1\ mm^2$
r	Radius of the motor	$0.01 \leq r \leq 0.1\ m$
t	Thickness of the motor	$0.0005 \leq t \leq 0.1\ m$
I	Current drawn by the motor	$0.1 \leq I \leq 6.0$ Amp
L	Stack length	$0.0566 \leq L \leq 10\ cm$

have an increasing trend over time, which indicates customers pay more attention to the factor as time passes. For the remaining sections, only *data set 1* is used for discussion except for the comparative study in Section 5.3.3.

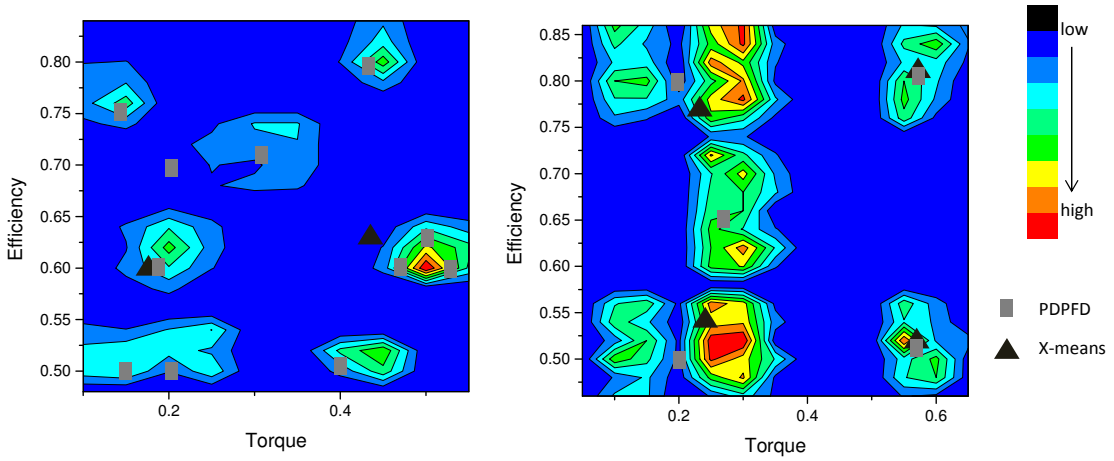


Figure 5.7: New orders in data set 1 (left) and data set 2 (right)

5.3.2 Profit Modeling

Two key factors (torque and efficiency) were assumed to be identified for the estimation of discounted price and cost functions. The discounted price and cost functions at one-step ahead (i.e., six months later) were formulated in Equation (5.23) and (5.24):

$$\hat{P}^*_{n+h} = \beta_{0(n+1)} + \beta_{1(n+1)} \sum_{i=1}^{k} a_{1i} + \beta_{2(n+1)} \sum_{i=1}^{k} a_{2i} + \beta_{3(n+1)} d \tag{5.23}$$

$$\hat{C}_{n+1} = \gamma_{0(n+1)} + \gamma_{1(n+1)} \sum_{i=1}^{k} a_{1i} + \gamma_{2(n+1)} \sum_{i=1}^{k} a_{2i} + \gamma_{3(n+1)} k \tag{5.24}$$

where a_1 is the torque; a_2 is the efficiency of a universal motor; d is the deviation in Equation (5.20); k is the number of product architectures. Since the demand (D_{n+1}) is given as the customers' new orders, the profit model at time $n+1$ is formulated by Equation (5.15). In order to maximize the profit, both the deviation and the number of architectures should be minimized. However, these two components are conflicting each other. When the number of architectures is increased, the deviation is decreased accordingly or vice versa. Both Equation(5.23) and (5.24) use the constant impact of the deviation and the number of product architectures.

Table 5.3 shows the historical regression coefficients of the discounted price fitted for historical data. The exponential smoothing based on innovations state space models was applied to model each time series (e.g., Torque from t=1 to t=12) using the *forecast* package [127] in R [108]. The mean column of Table 5.4 contains the point estimation of one-step-ahead prediction (i.e., t=13). The automatic forecasting method in Section 5.2.3 provided required parameters and initial states. Table 5.4 also shows lower (i.e., lo80 and lo95) and higher (i.e., hi80 and hi95) bounds of 80 and 95% prediction intervals based on the simulation method in Section 5.2.3. Instead of having the assumption of normally distributed errors, re-sampled errors or bootstrapping techniques were used to simulate future values. The cost model at t=13 is provided in the right side of Table 5.4.

Table 5.3: History of regression coefficients for discounted price

	t=1	t=2	t=3	t=4	t=5	t=6	t=7
Torque	34.99	34.50	34.20	34.00	33.50	33.09	32.79
Efficiency	22.01	22.49	22.8	23.00	23.50	23.60	23.60
Deviation	-18.00	-18.10	-18.20	-18.30	-18.50	-18.70	-19.10
Intercept	-0.0077	0	0	0	-0.0002	-0.0002	-0.0005
	t=8	t=9	t=10	t=11	t=12		
Torque	32.70	32.49	32.19	31.79	31.19		
Efficiency	23.60	23.80	24.79	25.49	26.29		
Deviation	-19.10	-19.29	-19.49	-19.69	-19.89		
Intercept	-0.0003	0	0.0001	0.0014	0.0001		

Table 5.4: Regression coefficients for discounted price and cost at t=13

for discounted price	mean	lo80	hi80	lo95	hi95	for cost	mean
Torque	30.86	30.68	31.03	30.59	31.13	Torque	26.0
Efficiency	27.07	26.63	27.50	26.40	27.73	Efficiency	24.8
Deviation	-20.10	-20.14	-20.06	-20.16	-20.04	k	2.5
Intercept	0.00027	-0.00262	0.00316	-0.00414	0.00469	Intercept	0

5.3.3 Individual Product Design Stage

Enterprise level

It was assumed that the maximum number of architectures was determined as 15 based on the manufacturer's capability and production environment. Positions of product architectures that minimize deviation errors for the one million new orders were identified using the k-means algorithm in Weka [107]. Since the k-means algorithm is the local optimizer, multiple seed values (10 different values) were used to get the k best clusters. Figure 5.7 shows the result with k=11 (left) and k=5 (right).

The profit model in Equation (5.15) at $n+1$ (i.e., $t = 13$) is now available. By utilizing Equations (5.6), (5.12) and (5.13), profits for mean, 80%, and 95% prediction intervals can be calculated as shown in Table 5.5. The top 4 ks were selected according to their profits. Though the selection of k is dependent on designers, the important fact is that the prediction intervals give the uncertainties of the predicted profit model. For example, T_{11} can have the profit range from 0.47 to 7.59 million dollars while T_{15} can have the range from -1.16 to 8.48 million dollars with a 80% prediction interval. It was assumed that the designer chose 11 architectures (T_{11}) with the expected profit of 4.03 million dollars. Then, the target T_{11} in Figure 5.7 was cascaded down to the engineering level.

Table 5.5: Architecture rankings based on prediction intervals of profit

		mean	lo80	hi80	lo95	hi95
Rank	The best	11/4.03	11/0.47	15/8.48	5/-1.01	15/11.04
(k/profit($ MM))	Second	15/3.66	5/-0.15	11/7.59	11/-1.41	11/9.47
	Third	13/2.68	7/-0.27	14/6.80	4/-1.48	14/9.14
	Fourth	14/2.40	6/-0.72	13/6.79	7/-1.48	13/8.89

* k is the number of architectures

Engineering level

The local objective function f_k (from Equation (5.21)) in this case study is the mass function of a universal motor. A mathematical universal motor model [3] is used as the analysis model $r(x)$ in Equation (5.21). Therefore, the objective function is to minimize the mass of motors and deviations between the target T_{11} and the response R while satisfying design constraints in Table 5.6.

Table 5.6: Design constraints for universal motors

Name	Constraint
Magnetizing intensity, H	H $\leq$ 5000Amp $\cdot$ turns/m
Feasible geometry	t<r
Power, P	P=300W
Mass, M	M $\leq$ 2.0kg

The Generalized Reduced Gradient (GRG) algorithm in Excel was used to solve this problem. Table 5.7 shows the engineering level optimization result with T_{11} from Figure 5.7 (i.e., T and η column).

Table 5.7: Universal motor specifications and performance responses

Motor	Product specifications (design variables)								Responses			
no.	N_c	N_s	$A_{wf}(mm^2)$	$A_{wa}(mm^2)$	I(Amp)	r(cm)	t(mm)	L(cm)	T(Nm)	η(%)	P(W)	M(kg)
1	998	105	0.476	0.347	3.72	3.05	2.73	2.34	0.30	70	300	0.984
2	998	105	0.430	0.416	5.25	4.91	2.44	1.69	0.40	49.8	300	0.809
3	998	105	0.431	0.467	3.81	4.57	2.41	1.58	0.20	68.5	300	0.637
4	997	36	0.149	0.149	5.22	1.47	1.47	1.88	0.10	50.0	300	0.218
5	997	75	0.346	0.346	4.24	2.51	2.51	4.49	0.49	61.6	300	1.294
6	997	101	0.560	0.560	3.29	2.62	2.62	4.50	0.41	79.4	300	1.821
7	995	61	0.255	0.255	3.47	1.67	1.67	2.26	0.10	75.3	300	0.406
8	995	72	0.335	0.334	4.41	2.49	2.49	4.46	0.50	59.3	300	1.252
9	995	53	0.213	0.213	4.35	1.82	1.82	2.63	0.17	60.0	300	0.443
10	995	45	0.199	0.199	5.15	1.82	1.82	2.62	0.2	50.7	300	0.422
11	995	70	0.319	0.319	4.41	2.42	2.42	4.23	0.45	59.3	300	1.126

Comparative study

As shown in the previous sections, the PDPFD algorithm combines the data-driven and market-driven approaches together for the target setting of the individual product design stage. In this section, PDPFD and a previous data-driven approach [64] are compared to validate the performance of the proposed algorithm.

The pure clustering based data-driven method [64] used the X-means clustering algorithm [147] to design aerodynamic particle separators. Out of 1000 data points, the X-means clustering found five cluster centroids (i.e., architectures) based on the Bayesian information criterion (BIC) [147]. From these five architectures (with the maximum BIC score), five product variants could be realized. However, the data-driven method calculated the production cost after determining the five product architectures. In contrast, the PDPFD algorithm considers the expected profit while simultaneously determining the product architectures. By design, PDPFD can generate profits that are equal to or greater than profits from the data-driven only method while BIC scores can be reduced.

Data set 1, 2, 3 were utilized for this comparative study. The X-means clustering algorithm in Weka [107] was used with the minimum (2) and maximum (15) number of architectures. Table 5.8 shows both results from PDPFD and the data-driven method. When PDPFD generated more architectures, the averages of within cluster sum of squared errors (SSE) were lower than that of the data-driven method. The data-driven method generated lower expected profit at the end because it maximized the BIC score first and then the profit was calculated sequentially with the determined number of product architectures. The PDPFD algorithm explored all the k values (e.g., k=2 to 15) and determined the best one by comparing profits. This shows the necessity of data-driven and market-driven combined approaches in product family architecture design as introduced in Figure 5.1 and 5.2.

Table 5.8: Result of comparative study

data set 1	k	Average SSE	BIC	Cost($ MM))/k	Revenue($ MM))/k	Expected profit($ MM))
PDPFD	11	0.003	-933	25.76	26.12	4.03
Data-driven method	2	0.104	-907	25.17	24.83	-1.57
data set 2	k	Average SSE	BIC	Cost($ MM))/k	Revenue($ MM))/k	Expected profit($ MM))
PDPFD	5	0.021	-934	28.56	28.65	0.42
Data-driven method	4	0.032	-795	27.00	26.71	-1.10
data set 3	k	Average SSE	BIC	Cost($ MM))/k	Revenue($ MM))/k	Expected profit($ MM))
PDPFD	2	0.096	-781	27.50	124.50	0.194
Data-driven method	4	0.025	-716	30.25	78.44	0.192

* k is the number of architectures

5.3.4 Product Family Design Stage

Based on the result (i.e., 11 motors) from the individual product design stage, the EM clustering in Weka [107] was applied to find clusters within design variables. Two clusters were identified for the number of wire turns (N_s) and the radius of the motor (r), and all other design variables had one cluster. Next, common parameters and scaling variables are selected. Based on Simpson et al. [3], the radius of the motor (r) and the thickness of the stator (t) were selected as the common parameters. Then, the engineering level optimization problem was resolved with respect to the six free design variables with the two fixed common parameters (i.e., r and t). Table 5.9 shows the result of the optimization problem which indicates two different platforms based on r and t (i.e., 4.74/2.21 and 2.21/2.21) shared by motors. The average weight of the motor family was increased by 30.2% (from 0.86kg to 1.12kg) but all weight constrains were satisfied (i.e., less than 2kg).

Table 5.9: Universal motor family design with fixed r and t

Motor	Product specifications (design variables)								Responses			
no.	N_c	N_s	$A_{wf}(mm^2)$	$A_{wa}(mm^2)$	I(Amp)	r(cm)	t(mm)	L(cm)	T(Nm)	η(%)	P(W)	M(kg)
1	1229	54	0.195	0.195	5.21	2.21	2.21	0.75	0.30	70	300	1.003
2	1227	116	0.475	0.475	5.25	4.74	2.21	1.31	0.40	49.8	300	1.975
3	1196	159	0.562	0.562	3.81	4.74	2.21	0.93	0.20	68.5	300	1.999
4	1437	54	0.220	0.220	5.21	2.21	2.21	0.64	0.10	50.0	300	0.346
5	1437	67	0.412	0.411	4.23	2.21	2.21	3.86	0.49	61.6	300	1.329
6	1050	86	0.597	0.597	3.29	2.21	2.21	5.69	0.41	79.4	300	1.894
7	1050	81	0.279	0.278	3.47	2.21	2.21	1.32	0.10	75.3	300	0.462
8	1050	64	0.354	0.353	4.41	2.21	2.21	5.18	0.50	59.3	300	1.262
9	1050	65	0.223	0.222	4.35	2.21	2.21	1.78	0.17	60.0	300	0.467
10	1050	55	0.208	0.207	5.15	2.21	2.21	1.77	0.2	50.7	300	0.443
11	1050	64	0.333	0.333	4.41	2.21	2.21	4.66	0.45	59.3	300	1.128

5.4 Conclusion

This chapter addresses how to determine optimal product family architectures with large-scale customer preference data. The proposed model expands clustering based data-driven approaches to incorporate a market-driven approach. The market-driven approach provides a profit model in the near future to determine the optimal position and number of product architectures among product architecture candidates generated by the k-means clustering algorithm. Unlike discrete choice analysis models which were used in previous market-driven approaches, a market value prediction method is proposed as a dynamic model which can capture and reflect the trend of customer preferences. Prediction intervals also provide market uncertainties of the dynamic profit model for product family architecture design.

The predictive, data-driven product family design (PDPFD) framework consists of the individual product design stage and the product family design stage. The individual design stage is a bi-level optimization model. At the enterprise level, a price prediction formulation is suggested with regression coefficients and time series modeling of the coefficients using exponential smoothing based on innovations state space models. In comparison with the RW model, the proposed model not only showed the better prediction accuracy for data with complex patterns but also provided prediction intervals which represent the dynamics and uncertainties of customer preferences. The k-means clustering algorithm is suggested to capture the effect of deviations between product architectures and customer requirements. Then, the optimal position and number of product architectures can be determined to maximize the profit model without pre-defined market segment information. With this market-driven target, the engineering level optimization problem is formulated to find designs which minimize deviations from the target. The next stage is the product family design stage where the EM clustering algorithm is applied to find clusters in common parameters so that the possibility of multiple platforms can be explored. Finally, the engineering level optimization is resolved with reduced design variables and common parameters. The example of universal electric motors design demonstrates the implementation of the proposed framework with large-scale data. The comparative study shows that the PDPFD algorithm can generate more profit than pure clustering based data-driven models, which shows the necessity of data-driven and market-driven combined approaches.

The PDPFD algorithm starts with a maximum number of architectures which is mainly dependent on the manufacturer's capability and production condition. If k is too big, then processing time for the algorithm will be very long. More efficient and data-driven ways should be explored to find the lower and upper bound of the number of k in the future. Furthermore, throughout this study, a scenario with no competition was used. It will be interesting to consider competing products for market-driven target setting as possible future work.

The next chapter will discuss a new usage modeling technique for life cycle assessment of systems. Since policy makers and manufacturers mainly focus on environmental impacts in the future, predictive design analytics can help the estimation more accurately. Preprocessing and time series modeling of sensor data will be discussed.

Chapter 6

Predictive Usage Mining for Life Cycle Assessment

In this chapter[1], the usage modeling technique, predictive usage mining for life cycle assessment (PUMLCA) algorithm, is proposed as an alternative of the conventional constant rate method. By modeling usage patterns as trend, seasonality, and level from a time series of usage information, predictive LCA can be conducted in a real time horizon, which can provide more accurate estimation of environmental impact. Large-scale sensor data of product operation is suggested as a source of data for the proposed method to mine usage patterns and build a usage model for LCA. The PUMLCA algorithm can provide a similar level of prediction accuracy to the constant rate method when data is constant, and the higher prediction accuracy when data has complex patterns. In order to mine important usage patterns more effectively, a new automatic segmentation algorithm is developed based on change point analysis. The PUMLCA algorithm can also handle missing and abnormal values from large-scale sensor data, identify seasonality, and formulate predictive LCA equations for current and new machines. Finally, the LCA of agricultural machinery demonstrates the proposed approach and highlights its benefits and limitations.

6.1 Introduction

The usage modeling in life cycle assessment (LCA) is rarely discussed despite the magnitude of environmental impact from the usage stage. In this study, a new perspective of dynamic LCA is proposed to consider time in LCA, especially the modeling of the usage stage. Among the life cycle stages of a product, the manufacturing stage, which is the chosen stage in the majority of LCA studies, can be considered as a one-time event, i.e., time-independent event. Although a dynamic inventory approach [73] attempted to relax this (e.g., the impact from material x or process y can be changed over time), the inventory data is considered constant in this study. On the other hand, the usage stage (with maintenance and end-of-life stages) is a time-dependent event, which means the lifespan of a product has a large impact on LCA. Many studies showed that the majority of environmental impact can come from the usage stage over life cycle (e.g., more than 60% for cars [74], more than 80% for off-load machinery (product of interest in this study) [75], and 80$\sim$90% for some small electronics [150]). Therefore, *how to model the usage stage in LCA* is critical and one of the

[1] Presented in [148] and submitted to [149].

main questions of this work.

LCA studies in literature usually utilized a constant rate [77, 78, 75, 1, 79] of usage information (hereinafter constant rate method) with the implicit assumption of steady-state processes. This method is simple and easy to apply, but if data has complex patterns (e.g., trend, seasonality and segments), the prediction accuracy of the constant rate method can be significantly reduced. Figure 6.1 shows the expected result from both the proposed model and the constant rate method. Based on the available historical data, a usage (e.g., diesel fuel consumption) model should be built and used for predicting the future usage profile. It can be seen that the constant rate method can misinterpret the upcoming usage profile.

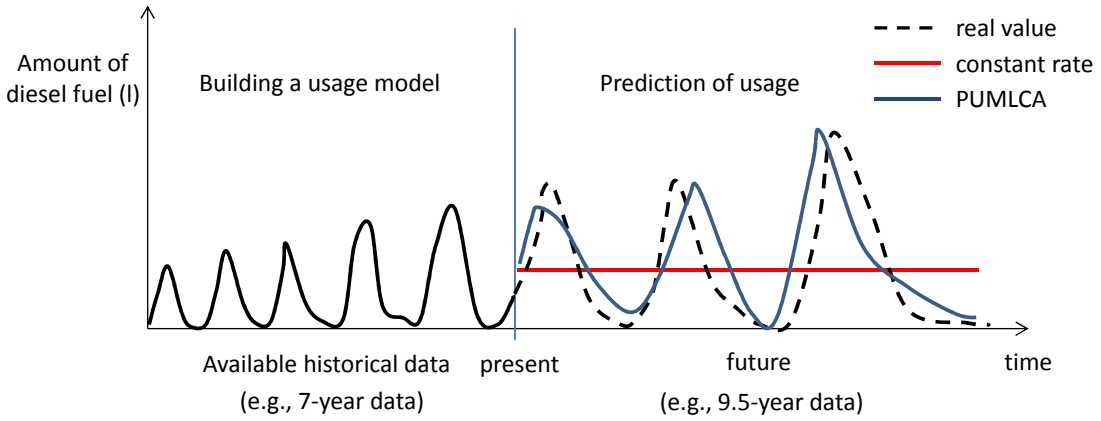


Figure 6.1: A prediction scenario of PUMLCA and constant rate method

One exception is Telenko and Seepersad [150] who proposed a usage context modeling technique in LCA using Bayesian network models. The usage context includes human, situational, and product variables. Based on a predefined probabilistic network of relevant usage patterns (e.g., weather$\rightarrow$ usage of electric kettle with probability of x), a usage profile and its variability can be modeled as a form of distribution. However, in order to apply this approach, causal relationships among different usage contexts should be known, which is expressed as a probabilistic network. For example, the usage of agricultural machinery (e.g., crop sprayer, harvester, nutrient applicator, etc.) can be affected by a various usage context (e.g., weather, soil, experience of farmers, price of fuel and crops, machine deterioration). It will be difficult to correlate these variables with specific usage information (e.g., diesel fuel consumption and operating hours). Furthermore, Telenko and Seepersad [150] did not consider time in LCA.

Alternatively, this study proposes a time series usage modeling technique, predictive usage mining for life cycle assessment (PUMLCA), as shown in Figure 6.2. Companies such as Caterpillar (PRODUCT Link^{TM}) and John Deere (JDLink^{TM}) have developed telematics systems for their machinery and have been gathering operational data in real time for various purposes: asset utilization monitoring, location tracking, fleet management, machine health prognos-

tics, etc. These large-scale time-stamped data sets are the sources of data for the PUMLCA algorithm. Usually, the whole picture of a usage profile is not available for currently deploying machines or new machines. Based on the limited past information, future usage patterns should be predicted for LCA as shown in Figure 6.1. Time series analysis is useful when future values should be predicted while explanatory variables are difficult to identify. By modeling time series usage information, not only can future usage patterns be obtained, but also variability (i.e., prediction interval). For example, Ma et al. [81, 115] showed that a trend of valuable information (demand and price) could be mined and reflected in system design using the combination of time series analysis and data mining.

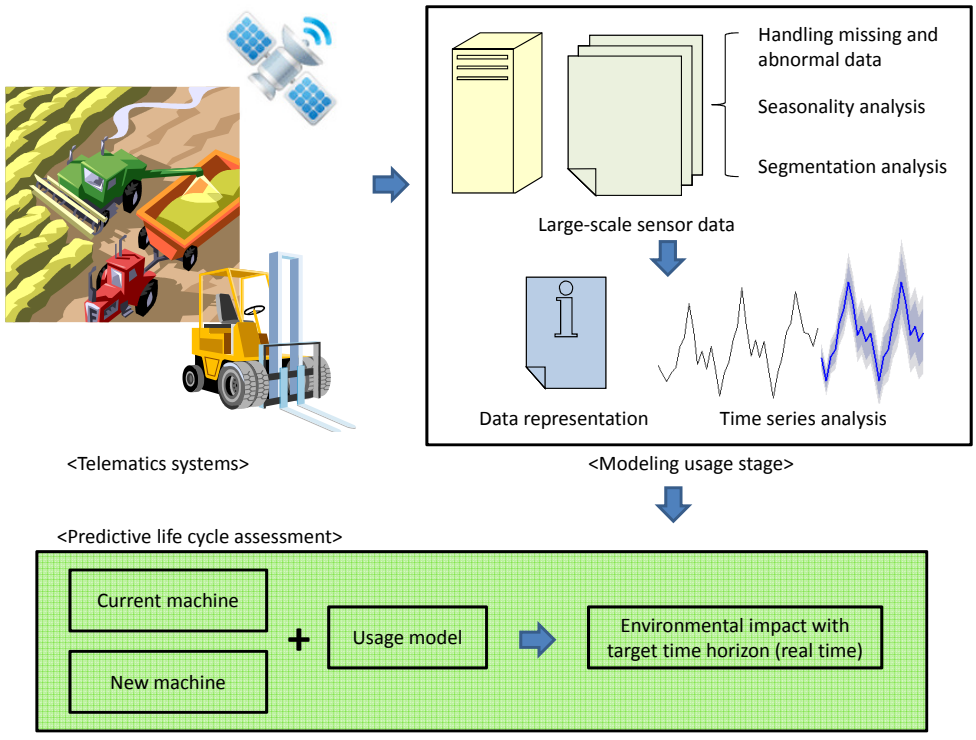


Figure 6.2: Overview of PUMLCA

Time series usage information, however, frequently shows highly seasonal activity periods with periodic no-activity or very low-activity periods. For example, combine harvesters are mainly operated during the harvest season with almost zero usage during the off-season. A similar pattern can be observed from seasonally used machinery. This pattern is also widespread for time series data of highly seasonal items such as Christmas, Easter and Halloween products. When analyzing and modeling this kind of time series data, a segmentation can help to find usage patterns more clearly by grouping distinct periods (e.g., off-season period) [151]. Segmentation algorithms [4] were proposed for various applications such as voice recognition, handwriting recognition, clustering, classification, etc. However, not much has been reported in the LCA literature whether segmentation algorithms can improve predictive capability.

Figure 6.3 shows the example. The usual time series segmentation (A (Electrocardiogram) in the figure, piecewise linear representation) is used for the approximation of a time series but the proposed segmentation (B (Monthly sales for a souvenir shop in Queensland, Australia) in the figure, dotted lines for predicted values) is designed to improve the predictive capability of time series modeling by grouping distinct periods and magnifying important patterns (e.g., ① '+' and '−' segments are separated and predicted, ② segments are regrouped). Therefore, *how to segment a time series for better LCA results* is another main question of this work.

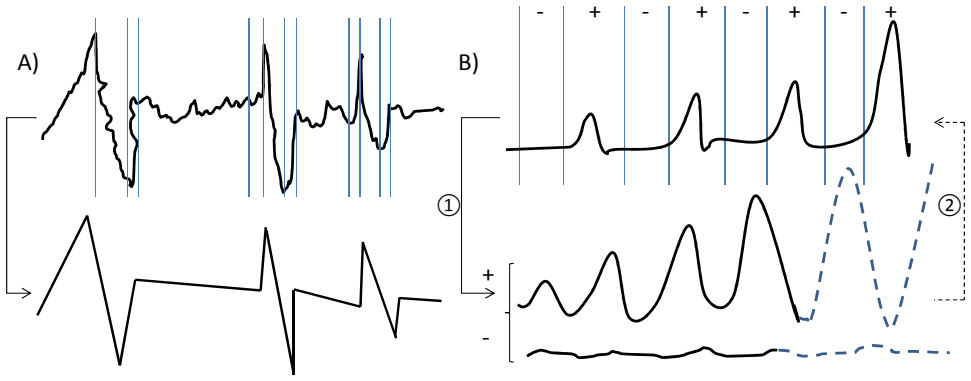


Figure 6.3: Time series segmentation A) piecewise linear representation (redrawn from [4]) B) segmentation for prediction (redrawn from [5])

The main contribution of this chapter is to propose the usage modeling technique, predictive usage mining for life cycle assessment (PUMLCA) algorithm, which enables predictive LCA in a real time horizon. The PUMLCA algorithm can provide a similar level of prediction accuracy to the constant rate method when data is constant, and a higher prediction accuracy when data has complex patterns. In order to mine important usage patterns (trend, seasonality and level) effectively from a time series, a new automatic segmentation algorithm is developed based on change point analysis. The PUMLCA algorithm can also handle missing and abnormal values from large-scale sensor data, identify seasonality, and formulate predictive LCA equations for current and new machines. Finally, the LCA of agricultural machinery demonstrates the proposed approach and highlights its benefits and limitations.

The rest of the chapter is organized as follows: Section 6.2 describes the PUMLCA algorithm. Section 6.3 provides design problems for current and new machines. Numerical prediction tests are presented for PUMLCA and the constant rate method in Section 6.4 followed by a case study of agricultural machinery in Section 6.5. The benefits and limitations of the proposed methodology along with future research directions are discussed in Section 6.6.

6.2 Methodology

Figure 6.4 outlines the predictive usage mining for life cycle assessment (PUMLCA) algorithm. There are five stages: data preprocessing for handling missing and abnormal values, seasonal period analysis, segmentation analysis, time series analysis, and predictive LCA. Details are explained in each subsection respectively. The algorithm starts from gathering time-stamped sensor data sets with usage information of interest. The amount of fuel (or energy) consumption and operating hours by work modes (e.g., idling and non-idling) are selected as the usage information. In this study, the usage information is viewed as a result of interactions among human, situational and product variables which are the components of the usage context [150]. For example, the amount of fuel consumed by work modes can be affected by user experience and preference (human variables), weather and soil (situational variables), and machine deterioration and efficiency (product variables). The patterns of the usage information (usage patterns) are defined as trend, seasonality and level in historical time series data. A trend is a long-term increase or decrease pattern; a seasonality is a repeated pattern with a fixed and known period; and a level is base values after removing trend and seasonality. Since a level can be considered as an initial value with a series of random errors, trend and seasonality are the two main patterns that will be mined.

6.2.1 Data Preprocessing

After collecting a time series of usage information of interest, it should be checked whether there are missing or abnormal values. Though it is assumed that the error rate of sensor data is very low and the incompleteness of data happens at random, it is still possible to have missing or abnormal values. In order to handle missing values (usually indicated as not available), various imputation techniques are available: 1) removing the missing values, 2) replacing the missing values with random values, adjacent values, mean or median, and 3) replacing the missing values based on values of a correlated variable. Since the volume of collected data is very large, any aforementioned method can be applied.

Unlike missing values, abnormal values (or outliers) are difficult to define. However, similar to the case of missing values, it is assumed that the sample size of abnormal values is much smaller than the volume of the original data and abnormal values are not generated systematically. There are two approaches: 1) three-sigma rule and 2) boxplot. The three-sigma rule states that approximately 99.73% of values lie within three standard deviations of the mean in Gaussian distribution. In other words, if the collected values (y_t) are considered random variables following the Gaussian distribution, abnormal values can be defined as values located outside of Equation (6.1):

$$\mu - 3\sigma \leq y_t \leq \mu + 3\sigma \tag{6.1}$$

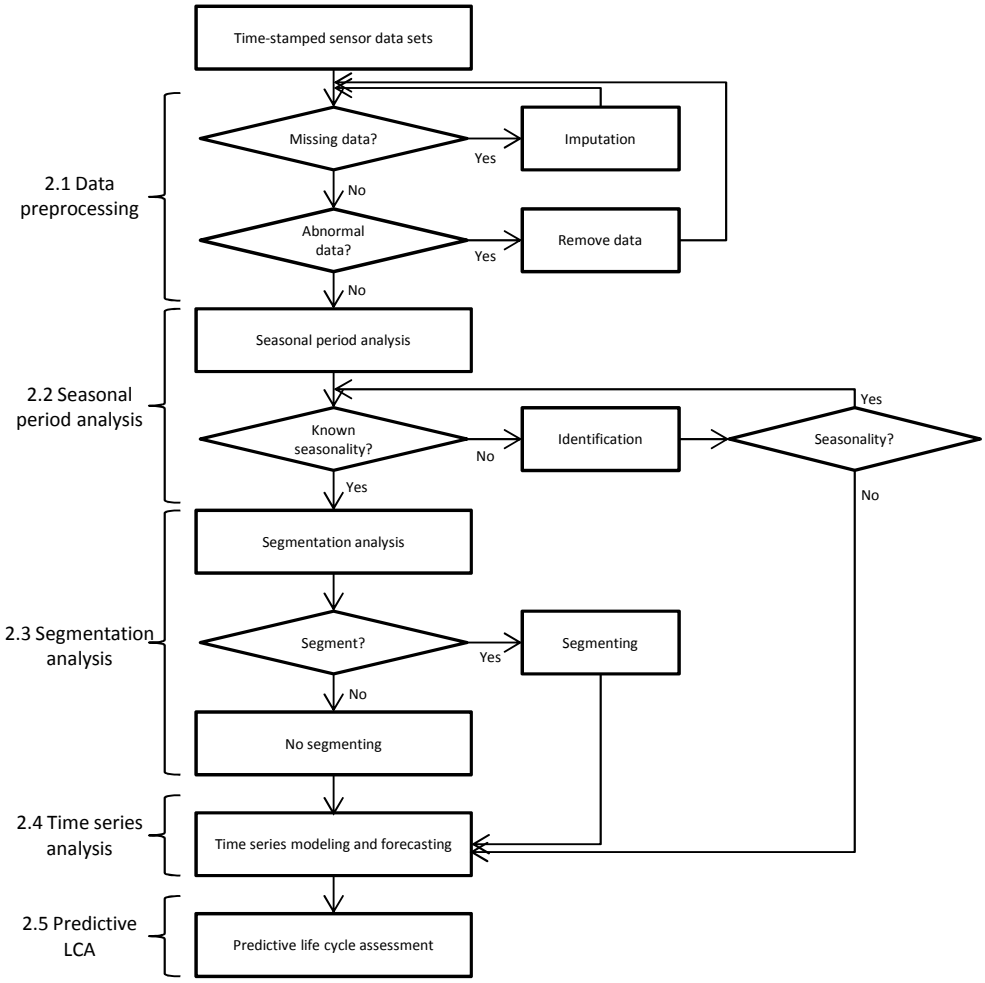


Figure 6.4: Overall framework of PUMLCA

where μ is the mean and σ is the standard deviation.

Another method to detect abnormal values is a boxplot. Abnormal values are defined as values located outside of Equation (6.2):

$$Q_1 - 1.5\text{IQR} \leq y_t \leq Q_3 + 1.5\text{IQR} \tag{6.2}$$

where Q_1 is the 25th percentile, Q_2 is the median or 50th percentile, Q_3 is the 75th percentile, and IQR refers to the interquartile range ($Q_3 - Q_1$). If data is distributed as the Gaussian distribution, Equation (6.2) can be expressed as $\mu \pm 2.7\sigma$. Figure 6.4 indicates that detected abnormal values are removed and handled by techniques for missing values.

6.2.2 Seasonal Period Analysis

The next step is to determine whether there are seasonal patterns, and if there is, what the length (period) of seasonality is. It should be noted that seasonality modeling will be conducted in Section 6.2.4, but without the information of the seasonal period, seasonality cannot be modeled. Examples of typical periods include 24 for an hourly series, 7 for a weekly series, 12 for a monthly series, 4 for a quarterly series, etc. If a seasonal period is known, the information can be used. If it is not known, then a dominant period should be identified with different seasonal representations of the original sensor data.

A periodogram [152] is suggested to identify the important seasonal period. The periodogram is a plot with the x-axis of frequencies and the y-axis of periodogram values. In order to derive the relationship between periodogram values and frequencies, it starts from the fact that a sum of cosine or sine waves can express a time series. For example, a cosine wave is given in Equation (6.3):

$$A cos(2\pi\omega t + \phi) = \beta_1 cos(2\pi\omega t) + \beta_2 sin(2\pi\omega t) \tag{6.3}$$

where A is an amplitude, ω is a frequency and ϕ is a phase. The equality is based on a trigonometric identity with $\beta_1 = A cos(\phi)$ and $\beta_2 = -A sin(\phi)$.

Then, a time series with n discrete time points is represented as a generalization of Equation (6.3), which is given by [152]:

$$y_t = \sum_{j=1}^{n/2} [\beta_1(\frac{j}{n}) cos(2\pi(\frac{j}{n})t) + \beta_2(\frac{j}{n}) sin(2\pi(\frac{j}{n})t)] \tag{6.4}$$

where (j/n) are frequencies ω_j (j cycles in n time points) for $j = 1, 2, \cdots, n/2$.

Now, the periodogram values are defined as [152]:

$$\begin{aligned} P(\frac{j}{n}) &= \hat{\beta}_1^2(\frac{j}{n}) + \hat{\beta}_2^2(\frac{j}{n}) \\ &= [\frac{2}{n}\sum_{t=1}^{n} y_t cos(2\pi(\frac{j}{n})t)]^2 + [\frac{2}{n}\sum_{t=1}^{n} y_t sin(2\pi(\frac{j}{n})t)]^2 \end{aligned} \tag{6.5}$$

where $\hat{\beta}_1$ and $\hat{\beta}_2$ are considered regression parameters and can be derived using the least squares estimates (i.e., the second equality). The periodogram is a sample spectral density, which can give the relative importance of frequencies.

In order to obtain the periodogram values, a discrete Fourier transform (DFT) can be used, which is given by [152]:

$$
\begin{aligned}
d(\frac{j}{n}) &= \frac{1}{n^2}\sum_{t=1}^{n} y_t exp(-2\pi i(\frac{j}{n})t)) \\
&= \frac{1}{n^2}[\sum_{t=1}^{n} y_t cos(2\pi(\frac{j}{n})t)) - i\sum_{t=1}^{n} y_t sin(2\pi(\frac{j}{n})t))]
\end{aligned}
\tag{6.6}
$$

where i is an imaginary number and the second equality is from Euler's formula.

The squared $d(j/n)$ times $(4/n)$ gives the periodogram values:

$$
\frac{4}{n}|d(\frac{j}{n})|^2 = \frac{4}{n}[\frac{1}{n}(\sum_{t=1}^{n} y_t cos(2\pi(\frac{j}{n})t))^2 + \frac{1}{n}(\sum_{t=1}^{n} y_t sin(2\pi(\frac{j}{n})t))^2]. \tag{6.7}
$$

Equation (6.7) can be computed by a fast Fourier transform (FFT), which is available in many computing software platforms such as R and Matlab. R codes to plot a periodogram with FFT is provided by [152].

Finally, the dominant period (i.e., reciprocal of a frequency (j/n)) can be identified by satisfying:

$$
\arg\max_{j/n} P(j/n) \tag{6.8}
$$

One helpful treatment before plotting a periodogram is detrending time series usage information (i.e., remove a trend). Two possible methods of detrending will be presented in Section 6.2.4. Also, from a practical standpoint, users can limit a range of frequencies as a meaningful range by their definition.

6.2.3 Segmentation Analysis

There are two types of segmentation analysis: deterministic and automatic. Deterministic segmentation analysis can be used when some segments of given time series data show deterministic patterns, e.g., zero usages over time within specific periods. If this prior knowledge is not available or patterns are not deterministic with variable periods, automatic segmentation analysis should be applied. In this study, a new automatic segmentation algorithm with the change point analysis is presented.

Figure 6.5 shows the schematic of the automation segmentation algorithm. A period (n/j) is identified from Section 6.2.2 and the number of data points n is proportional to the period (i.e., $n/j = jp/j = p$). For each period, there are p time indexes, $m_1, m_2, \cdots, m_p$. For example, a period 12 has 12 time indexes which are January, February, $\cdots$, December. The goal of this algorithm is to find a shared segment (SS) over periods. sp_j denotes a segment, which is a set of p time indexes in the period p_j. The segment does not contain any change point.

Change point analysis is a statistical technique that can detect multiple change points within a time series [153]. When a discrete time series, $y_{1:n} = \{y_1, \cdots, y_n\}$, is given, positions of change points, $\tau_{1:m}$ ($\tau_0 = 0$ and $\tau_{m+1} = n$) can

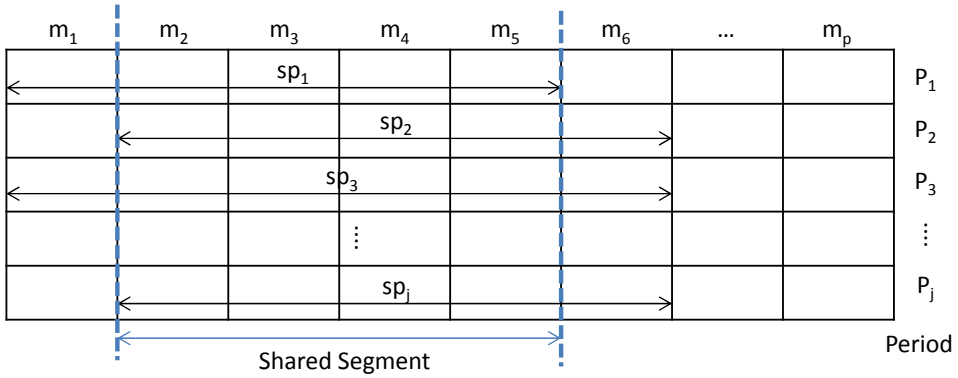


Figure 6.5: A schematic of automatic segmentation algorithm

be identified if the statistical properties of $y_{1:\tau_1}, y_{\tau_1+1:\tau_2}, \cdots, y_{\tau_m+1:n}$ are different in some sense. In this study, changes in mean are adopted, although changes in variance are another option. In order to identify change points, an objective function is given by [153]:

$$F(n) = \min_{\tau}\{\sum_{i=1}^{m+1}[C(y_{(\tau_{i-1}+1):\tau_i}) + \beta]\} \tag{6.9}$$

where C is a cost function for a segment and β is a penalty. For $t < n$, a recursive expression can be determined as follows [153] and solved in turn by dynamic programming:

$$\begin{aligned} F(n) &= \min_{t}\{\min_{\tau\in\tau_{1:t}}\sum_{i=1}^{m}[C(y_{(\tau_{i-1}+1):\tau_i} + \beta)] + C(y_{(t+1):n}) + \beta]\} \\ &= \min_{t}\{F(t) + C(y_{(t+1):n}) + \beta\} \end{aligned} \tag{6.10}$$

A pruned exact linear time (PELT) method [153] was proposed to solve Equation (6.10) more efficiently with a pruning procedure instead of searching all possible change points. During iterations for $t < s < n$, only a set of t satisfying Equation (6.11) will be considered:

$$F(t) + C(y_{(t+1):s}) + K \leq F(s) \tag{6.11}$$

where K is a constant.

As a cost function, the negative of maximum log-likelihood is used, which is given by [153]:

$$C(y_{(t+1):n}) = -\max_{\theta}\sum_{i=t+1}^{n} logf(y_i|\theta) \tag{6.12}$$

where $f(y_i|\theta)$ is a density function with the parameter θ for a segment.

As a penalty, there are some options such as Akaike's Information Criterion (AIC, $\beta = 2p$) and Bayesian Information Criterion (BIC, $\beta = plog(n)$), where p is the number of added parameters for a change point. It is also possible to specify a type I error (e.g., 0.05 or 0.01) as a penalty value using an asymptotic distribution [154]. The PELT algorithm is implemented in R [154].

The automatic segmentation algorithm based on the change point analysis (i.e., PELT) is provided in Algorithm 2. The goal of this algorithm is to find shared segments over seasonal periods which contain no change point. Unlike the PELT algorithm, change points will be identified within a seasonal period. A penalty, β, should be selected by users. As the penalty value increases, less change points will be identified and the algorithm will be less sensitive over close values. A segment is defined as a group of members within a seasonal period. At least two members are required to be a segment (e.g., $y_{1:3}$ in line 12). In line 4, τ^* contains the possible positions of change points, which are p time indexes within each period (e is the indexes of periods). In line 5$\sim$8 [153], the PELT algorithm is implemented with the pruning procedure in line 8. R_{τ^*} is the set of τ^*; τ' is the identified optimal position of change points; CP_e denotes the optimal positions of change points (τ^*) for each period, which is the result of the first part of the algorithm in line 10. Line 12 makes a set of segments, S_e, for each period based on the identified optimal change points (CP_e). Note that $\tau_{1:m_p} = (\tau_1, \cdots, \tau_{m_p})$. Line 13 finds shared segments (SS) over different periods. At this point, it is possible that change points can exist among the sets, S_e, in the shared segments, which indicates that those segments are not similar patterns that repeat periodically. Line 14 makes one new time series (NS) using shared segments of each period (e.g., SS_{p_1} represents the shared segment of the first period). Line 15 applies the PELT method for the new series with no period and a new change point set, CP$'$, is returned in Line 16. The output depends on the new change point set. If there is no change point, the shared segments and the remaining data are grouped as different time series. If there is a change point, no segmentation will be implemented.

Based on the result of the automatic segmentation algorithm, time series analysis methods in the next section will be applied to each segmented time series. Now, each time series has a new period, which is the number of seasonal time indexes.

6.2.4 Time Series Analysis

Time series analysis includes modeling time series data by extracting important patterns and forecasting future values from the fitted model. The two most widely used time series analysis techniques [5] are adopted in this study: exponential smoothing (ETS) and autoregressive integrated moving average (ARIMA). Since "each has its strengths and weaknesses" [127], either method can be selected by users. Observations are denoted by y_t and a forecast of h ahead time based on all the data up to time t is denoted by $\hat{y}_{t+h|t}$ where h is a real time horizon.

Algorithm 2 Automatic Segmentation

1: A time series, $y_{1:n}$ with n number of data points
2: A seasonal period, p, where p=n/j with j cycles
3: A measure of fit C$(\cdot)$ and a penalty β
4: **for** $\tau^* = 1, \cdots, m_p$ and $e = p_1, p_2, \cdots, p_j$ **do**
5: Calculate $F(\tau^*) = \underset{\tau \in R_{\tau^*}}{min}\,[\{F(\tau) + C(y_{(\tau+1):\tau^*}) + \beta\}$
6: Let $\tau' = \underset{\tau \in R_{\tau^*}}{\arg\min}[\{F(\tau) + C(y_{(\tau+1):\tau^*}) + \beta\}$
7: Set $CP_e(\tau^*) = [cp(\tau'), \tau']$
8: Set $R_{\tau^*+1} = \{\tau \in R_{\tau^*} \cup \{\tau^*\} : F(\tau) + C(y_{(\tau+1):\tau^*} + K \leq F(\tau^*))\}$
9: **end for**
10: **return** $CP_{p_1}, CP_{p_2}, \cdots, CP_{p_j}$
11: **for** $e = p_1, p_2, \cdots, p_j$ **do**
12: Set $S_e = \{y_{1:\tau_1^*}, y_{(\tau_1^*+1):\tau_2^*}, \cdots, y_{(\tau_{m_p-1}^*+1):\tau_{m_p}^*}\}$
13: Find $SS = \{S_{p_1} \cap S_{p_2} \cap \cdots \cap S_{p_j}\}$
14: Let $NS = \{SS_{p_1}, SS_{p_2}, \cdots, SS_{p_j}\}$
15: Apply line 4$\sim$9 to NS
16: Get CP$'(\tau^*)$
17: **end for**
18: **return**
19: **if** CP$'(\tau^*)$=null **then**
20: group SS as one time series and remaining as another time series
21: number of time series (s) = z
22: **else**
23: no segmentation, $s = 1$ (i.e., original data)
24: **end if**

Exponential Smoothing

The ETS models refer to an exponential smoothing family (e.g., simple exponential smoothing, Holt's linear trend model, Holt-Winters seasonal model, etc.) based on the innovations state space framework [96]. The ETS model identifies key components of a time series (trend and seasonality) and expresses their relationships (additive and multiplicative) using exponential smoothing.

The simplest model of ETS is given as:

$$\hat{y}_{t+1} = \hat{y}_t + \alpha(y_t - \hat{y}_t) \tag{6.13}$$

where α is a parameter between zero and one. Equation (6.13) represents that the new forecast is the combination of the old forecast and the error from the last forecast. Similar to Equation (6.13), there are 30 ETS models with a combination of trend (none, additive, additive damped, multiplicative and multiplicative damped), seasonality (none, additive and multiplicative) and error (additive and multiplicative) [96].

All the 30 ETS models can be expressed as innovations state space models and the general model is given as [96]:

$$y_t = w(x_{t-1}) + r(x_{t-1})\varepsilon_t \tag{6.14}$$

$$x_t = f(x_{t-1}) + g(x_{t-1})\varepsilon_t \tag{6.15}$$

where x_t is the state vector which contains unobserved components such as the level, trend, and seasonality of a time series; $w()$ and $r()$ are scalar functions; $f()$ and $g()$ are the vector functions; and ε_t is the white noise process with variance σ^2. The white noise process is a process that has zero mean, constant and finite variance, and uncorrelated series. Using this innovations state space framework, Hyndman et al. [96] showed that prediction interval can be obtained along with a point forecast.

In order to get a forecast, $\hat{y}_{t+h|t}$, a recursive expression was summarized as follows [96]:

$$\hat{y}_{t|t-1} = w(x_{t-1}) \tag{6.16}$$

$$\varepsilon_t = (y_t - \hat{y}_{t|t-1})/r(x_{t-1}) \tag{6.17}$$

$$x_t = f(x_{t-1}) + g(x_{t-1})\varepsilon_t \tag{6.18}$$

Then, a simulation approach [127] can be used to simulate ε_t for a forecast with a prediction interval.

The remaining part is the identification of trend and seasonality, which is called as the decomposition of a time series. First, the trend component can be estimated ($\hat{T}_t$) by a moving average smoothing. The moving average smoothing of order m is given by [5]:

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j} \tag{6.19}$$

where $m = 2k + 1$. The order of the moving average smoothing is a seasonal period, and if the seasonal period is not known, usually odd orders (e.g., 3, 5, 7, 9, etc.) can be applied [5]. A larger order gives a smoother fit. Then, detrended time series data can be obtained as $y_t - \hat{T}_t$ for the additive model or $y_t/\hat{T}_t$ for the multiplicative model. It should be noted that this is one method to obtain a detrended series for the seasonal period analysis in Section 6.2.2.

Second, the seasonal component can be estimated from detrended series data. An average of each seasonal time index over seasonal periods (e.g., all values in January for monthly data) gives the seasonal component, $\hat{S}_t$.

ARIMA

While the ETS model represents a time series as exponential smoothing of trend and seasonality, the ARIMA model is based on autocorrelations in the time series. The ARIMA model (without seasonality) is a combination of three models given as [5]:

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-B)^d y_t = c+(1+\theta B+\cdots+\theta_q B^q)e_t \tag{6.20}$$

where the first parenthesis is an autoregressive (AR) model of order p, the second parenthesis is an integration (or differencing operation), and the third parenthesis on the right-hand side is a moving average (MA) model of order q. B represents a backward shift operator, e.g., $By_t = y_{t-1}$.

The AR model of order p is given by:

$$y_t = c+\phi_1 y_{t-1}+\phi_2 y_{t-2}+\cdots+\phi_p y_{t-p}+e_t \tag{6.21}$$

where c is a constant and e_t is white noise. This is a linear combination of past observations.

The differencing operation of order 1 and order 2 is given as:

$$y'_t = y_t - y_{t-1} \tag{6.22}$$

$$y''_t = y'_t - y'_{t-1} \tag{6.23}$$

The determination of differencing can be made by statistical inference called unit root tests [5]. It should be noted that this is another method for detrending time series data for the seasonal period analysis in Section 6.2.2.

The MA model of order q is given as:

$$y_t = c+e_t+\theta_1 e_{t-1}+\theta_2 e_{t-2}+\cdots+\theta_q e_{t-q} \tag{6.24}$$

This is a linear combination of past forecast errors.

Finally, seasonal ARIMA model can be written as [5]:

$$\begin{aligned}(1-\phi_1 B-\cdots-\phi_p B^p)(1-\Phi_1 B^m-\cdots-\Phi_P B^{Pm})(1-B)^d(1-B^m)^D y_t \\ = c+(1+\theta_1 B+\cdots+\theta_q B^q)(1+\Theta_1 B^m+\cdots+\Theta_Q B^{Qm})e_t\end{aligned} \tag{6.25}$$

where lower-case letters p, d, and q are orders for non-seasonal AR, integration, and MA models; upper-case letters P, D, and Q are orders for seasonal AR, integration, and MA models; and m is a period.

In order to forecast future values based on a fitted ARIMA model, Equation (6.25) can be expanded so that only y_t will be shown on the left-hand side. By rewriting it as $\hat{y}_{t+h|t}$, a recursive expression can be solved for a forecast of h ahead time.

Close observation for both ETS and ARIMA models reveals similarities. The ETS model starts identifying trend and seasonality and the ARIMA model uses the differencing operation to remove trend and seasonality (i.e., stationarity). The ETS then expresses a series using past level, trend and seasonality with exponentially decreasing weights while the ARIMA expresses a series using past observations and forecast errors.

Automatic Modeling of ETS and ARIMA

As shown previously, the ETS and ARIMA require parameter estimation and model selection. Hyndman and Khandakar [127] provided an automatic forecasting algorithm to handle a large number of univariate time series data. The algorithm is implemented in R package *forecast*. This section briefly introduces the automatic forecasting algorithm for the ETS and ARIMA models.

The automatic forecasting algorithm for the ETS models can be summarized as follows: 1) apply all 30 models and optimize parameters of each model, 2) select the best model based on a penalized likelihood such as AIC and BIC, and 3) forecast future values and obtain prediction intervals based on the selected model.

The automatic forecasting algorithm for the ARIMA can be summarized as follows: 1) apply four possible models and select the best model based on a penalized likelihood, 2) apply 13 variations on the current model and repeat the process if a better model can be identified based on a penalized likelihood, and 3) forecast future values and obtain prediction intervals based on the selected model. Details of these algorithm can be found in the work of Hyndman and Khandakar [127].

6.2.5 Predictive Life Cycle Assessment

The difference between predictive LCA and original LCA is to model the usage stage (with maintenance and end-of-life stages) as a time series and to forecast future impact in a real time horizon. The total life cycle impact of a product can be expressed as [1]:

$$I^{total} = I^{mfg} + I^{usage} + I^{maint} + I^{eol} \tag{6.26}$$

where I^{mfg}, I^{usage}, I^{maint} and I^{eol} represent the impact of manufacturing, usage, maintenance, and end-of-life stage. In

the equation, a constant fuel (or energy) consumption rate in the usage stage and replacement cycles in the maintenance stage are components that are dependent upon the expected lifespan. However, the time in Equation (6.26) is nominal, e.g., 10 years instead of specifying a time horizon such as from October 2014 to December 2024.

Instead, Equation (6.27) gives the total environmental impact in a real time horizon:

$$\sum_{t=i}^{l} I^{total} = I^{mfg} + \sum_{t=i}^{l} [I_i^{usage} + I_i^{maint} + I^{eol}] \tag{6.27}$$

where l is the expected life time starting from time i. The impact of manufacturing can be considered as a one-time event while the impacts of usage, maintenance, and end-of-life are affected by time series usage information.

The impact of manufacturing is given as [1]:

$$I^{mfg} = \sum_{r} e_r^{raw} N_r + \sum_{p} e_p^{process} N_p + \sum_{s} e_s^{trans} N_s \tag{6.28}$$

where e_r^{raw}, $e_p^{process}$, and e_s^{trans} represent unit environmental impact of raw materials (r), manufacturing processes (p), and transportation (s); N_r, N_p, and N_s denote the number of units of raw materials, manufacturing processes, and transportation.

The impacts of usage, maintenance, and end-of-life are given as:

$$\sum_{t=i}^{l} I^{usage} = \sum_{t=i}^{l} I_i^{fuel} + \sum_{t=i}^{l} I^{emission} = \sum_{t=i}^{l} e^{fuel} N_{ft} + \sum_{q} \sum_{t=i}^{l} e_q^{emission} ER_q OH_t \tag{6.29}$$

$$\sum_{t=i}^{l} I^{maint} = \sum_{m} \sum_{t=i}^{l} e_m^{maint} N_m \left\lceil \frac{max(OH_t - RC_m, 0)}{RC_m} \right\rceil \tag{6.30}$$

$$\sum_{t=i}^{l} I^{eol} = e_{used}^{eol} + \sum_{m} \sum_{t=i}^{l} e_{replace}^{eol} N_m \left\lceil \frac{max(OH_t - RC_m, 0)}{RC_m} \right\rceil \tag{6.31}$$

where I^{fuel} and $I^{emission}$ are the impacts of fuel production as in Equation (6.28) and emissions while running an equipment; e^{fuel}, $e_q^{emission}$, e_m^{maint}, e_{used}^{eol}, and $e_{replace}^{eol}$ are the unit impacts of fuel, emissions, manufacturing of maintenance part m as in Equation (6.28), and end-of-life processing of a used product and replaced part (m); N_{ft} is the amount of fuel consumed per liter; N_m denotes the number of units of part m (in a product); ER_q is the emission rate of emissions q in g/hr; OH_t is the operating time in hours; RC_m is the replacement cycle of part m in hours; $\lceil \cdot \rceil$ is the ceiling function. The value of a ceiling function will give the number of replacements for part m. All the unit impacts can be obtained from the ecoinvent database (version 2.2), which is available in the LCA software SimaPro. Note that this study only considers energy-related impacts (e.g., fuel and electricity) of the usage stage. Other consumables are not considered, e.g., coffee and water for coffee machines, paper and ink for printers, etc.

Section 6.2 described the proposed algorithm from data preprocessing to predictive LCA formulation. Note that the algorithm starts from the available time-stamped data sets (top of Figure 6.4) and it is not discussed how many data sets should be available for the algorithm. Empirical studies show that if the available data is not enough to identify useful patterns (e.g., only a few data points), then the result from Section 6.2.4 is identical with the constant rate method, which is smoothing by averaging available data points. Actually, the constant rate method can be considered as a special case of the proposed time series analysis methods. In the next section, the proposed LCA formulation will be elaborated with design problems.

6.3 Design Problems with PUMLCA

Two system design cases are considered in this study, which is shown in Figure 6.6. The first case, analysis for sustainability, is when current machines need to be analyzed for sustainability. In this case, enough usage data is available with manufacturing, maintenance and end-of-life data. Life cycle information includes all the information from life cycle stages and the expected lifespan or target time horizon.

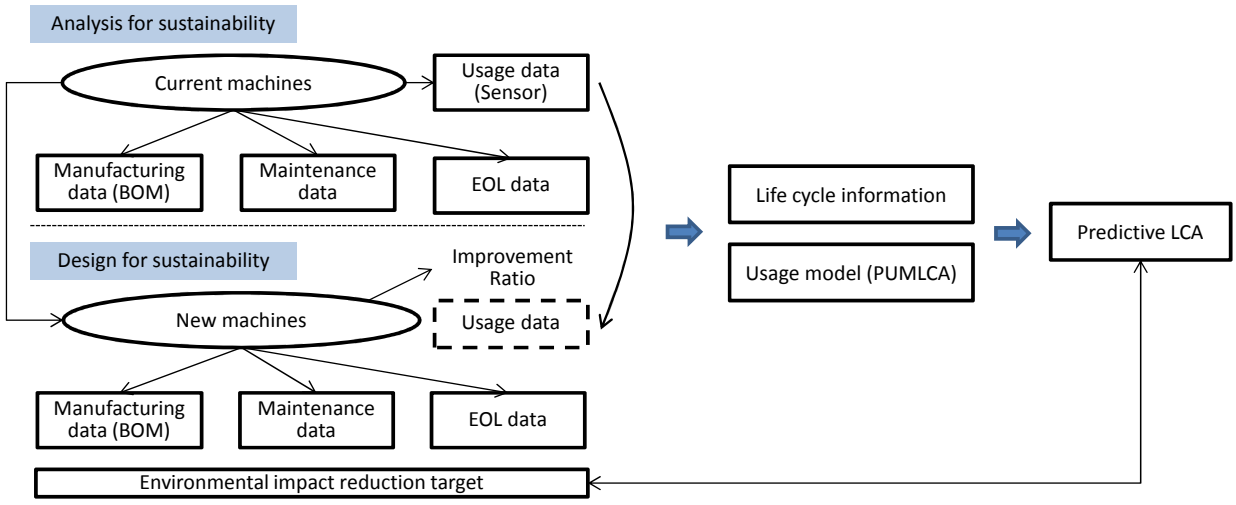


Figure 6.6: Two system design cases for predictive LCA

The amount of fuel consumed, N_{ft}, and operating hour, OH_t, are the time series usage information. The fitted models for N_{ft} and OH_t from ARIMA or ETS are $TS_{ts}^{N_f}$ and TS_{ts}^{OH} with the number of segments s in Algorithm 2. For example, $TS_{ts}^{N_f}$ can be either Equation (6.32) and (6.33), or Equation (6.34):

$$N_{fts} = w(x_{t-1}) + r(x_{t-1})\varepsilon_t \tag{6.32}$$

$$x_t = f(x_{t-1}) + g(x_{t-1})\varepsilon_t \tag{6.33}$$

$$\begin{aligned}(1-\phi_1 B - \cdots - \phi_p B^p)(1-\Phi_1 B^m - \cdots - \Phi_P B^{Pm})(1-B)^d (1-B^m)^D N_{fts} \\ = c + (1+\theta_1 B + \cdots + \theta_q B^q)(1+\Theta_1 B^m + \cdots + \Theta_Q B^{Qm}) e_t\end{aligned} \tag{6.34}$$

The environmental impact of current machines can be predicted as follows based on Equation (6.28), (6.29), (6.30) and (6.31):

$$I^{mfg} = \sum_r e_r^{raw} N_r + \sum_p e_p^{process} N_p + \sum_s e_s^{trans} N_s \tag{6.35}$$

$$\sum_{t=i}^{l} I^{usage} = \sum_{t=i}^{l}\sum_{s=1}^{z} e^{fuel} TS_{ts}^{N_f} + \sum_q \sum_{t=i}^{l}\sum_{s=1}^{z} e_q^{emission} ER_q TS_{ts}^{OH} \tag{6.36}$$

$$\sum_{t=i}^{l} I^{maint} = \sum_m \sum_{t=i}^{l}\sum_{s=1}^{z} e_m^{maint} N_m \left\lceil \frac{max(TS_{ts}^{OH} - RC_m, 0)}{RC_m} \right\rceil \tag{6.37}$$

$$\sum_{t=i}^{l} I^{eol} = e_{used}^{eol} + \sum_m \sum_{t=i}^{l}\sum_{s=1}^{z} e_{replace}^{eol} N_m \left\lceil \frac{max(TS_{ts}^{OH} - RC_m, 0)}{RC_m} \right\rceil \tag{6.38}$$

The second case, design for sustainability, is for the assessment of the new machines' sustainability when the target of environmental impact reduction should be applied to current machines due to new environmental regulations and enforcement. In this case, it is assumed that the new machines are upgraded versions of current machines. For example, new machines can improve the fuel efficiency with different materials or components. While these BOM (bill of materials) changes might increase the environmental impact of the manufacturing stage, the efficient fuel usage can reduce the environmental impact of the usage stage. As shown in Figure 6.6, the main difference between the current machines and new machines is the availability of usage data (or usage model). The proposed method for the estimation of usage information is to use the improvement ratio which is defined as follows:

$$\delta_{N_f} = \frac{(N_f / W_{unit})_{\text{new machine}}}{(N_f / W_{unit})_{\text{current machine}}} \tag{6.39}$$

$$\delta_{OH} = \frac{(OH / W_{unit})_{\text{new machine}}}{(OH / W_{unit})_{\text{current machine}}} \tag{6.40}$$

where δ_{N_f} is the improvement ratio for the amount of fuel consumption, δ_{OH} is the improvement ratio for the operating hours, and W_{unit} is a unit of work. For example, if a new nutrient applicator can apply fertilizers with high precision

and speed, these can be expressed as δ_{N_f} and δ_{OH} with the work unit of the square meter (m^2) from testing data. Then, the sensor data of current nutrient applicators can be used with the δ_{N_f} and δ_{OH} as follows for the environmental impact of the new machine:

$$I^{mfg} = \sum_r e_r^{raw} N_r + \sum_p e_p^{process} N_p + \sum_s e_s^{trans} N_s \tag{6.41}$$

$$\sum_{t=i}^{l} I^{usage} = \sum_{t=i}^{l} \sum_{s=1}^{z} e^{fuel} \delta_{N_f} TS_{ts}^{N_f} + \sum_q \sum_{t=i}^{l} \sum_{s=1}^{z} e_q^{emission} ER_q \delta_{OH} TS_{ts}^{OH} \tag{6.42}$$

$$\sum_{t=i}^{l} I^{maint} = \sum_m \sum_{t=i}^{l} \sum_{s=1}^{z} e_m^{maint} N_m \left\lceil \frac{max(\delta_{OH} TS_{ts}^{OH} - RC_m, 0)}{RC_m} \right\rceil \tag{6.43}$$

$$\sum_{t=i}^{l} I^{eol} = e_{used}^{eol} + \sum_m \sum_{t=i}^{l} \sum_{s=1}^{z} e_{replace}^{eol} N_m \left\lceil \frac{max(\delta_{OH} TS_{ts}^{OH} - RC_m, 0)}{RC_m} \right\rceil \tag{6.44}$$

The LCA result from Equation (6.41), (6.42), (6.43) and (6.44) estimates the environmental impact of the new machine. The result can also show whether the target of environmental impact reduction is satisfied. Otherwise, new design strategy should be explored. Note that the two design cases can be viewed as phases of a single design case, i.e., evaluation of current sustainability and redesign.

6.4 Numerical Prediction Tests for PUMLCA

In this section, a set of different data is tested to validate the prediction performance of PUMLCA. Due to the significance of environmental impact from the usage stage in LCA, the prediction accuracy of a time series usage model will play an important role for the estimation of environmental impact. The conventional method to model the usage stage is the constant rate method, which is the average of observations. The hypotheses are 1) the PUMLCA algorithm can provide a similar level of prediction accuracy to the constant rate method when data is constant with small random errors (i.e., steady-state processes), hereinafter *data 1*, 2) the PUMLCA can predict future values more accurately than the constant rate method when data has a trend, hereinafter *data 2*, 3) the automatic segmentation algorithm in PUMLCA can help to improve the predictive modeling when data has a trend and segments, hereinafter *data 3*, and 4) the PUMLCA algorithm can provide higher prediction accuracy than the constant rate method when prediction is required for specific periods within the whole prediction horizon.

Data sets (*data 1, 2, 3*) with monthly seasonal patterns were generated and the procedures are described in Section 4.1 for the hypotheses 1), 2) and 3). The three types of data sets were also used to test the hypothesis 4). In terms of the target of prediction, this study proposes to use not only the aggregated life cycle values (accuracy) but also the seasonal values of time series usage information (variance) because different time horizon scenarios can be tested. For

example, monthly usage data is used to predict the next two-year values and the accumulated two-year values can be used to assess the environmental impact of life cycle as an accuracy measure. If the environmental impact of next quarter or specific periods within two years is required to be estimated, the accuracy of the predicted seasonal values (i.e., monthly values) will determine the quality of the analysis, which can be considered a variance measure. This is related to the fourth hypothesis. Therefore, the best model should provide good predictions of both values: high accuracy (aggregated life cycle values) and low variance (seasonal values).

As a prediction performance measure, mean absolute percentage error (MAPE) and mean absolute error (MAE) were used. Equation (6.45) and (6.46) show MAPE and MAE with the predicted values, b_1, b_2, $\cdots$, b_m and the real values, d_1, d_2, $\cdots$, d_m. MAPE is scale-independent so that results from different data sets can be compared. However, by design, if the actual values are close to zero, MAPE cannot be defined. In this case, the scale-dependent measure, MAE, was used.

$$\text{Mean Absolute Percentage Error} = \frac{100(|\frac{b_1-d_1}{d_1}| + \cdots + |\frac{b_m-d_m}{d_m}|)}{m} \tag{6.45}$$

$$\text{Mean Absolute Error} = \frac{|b_1-d_1| + \cdots + |b_m-d_m|}{m} \tag{6.46}$$

Throughout the numerical tests, only positive values were accepted as valid values. Negative values were set to zero. In order to handle non-negative data, one common method is the Box-Cox transformations [5], which includes logarithms and power transformations. More theoretical discussions can be found in the literature [96].

6.4.1 Data generation

To test the first hypothesis, the following data generation procedure was applied: 1) a value from 100 to 1000 was randomly chosen using a random number generator for each month, 2) by adding a random error between -5 and 5 for each month, monthly data with seasonal patterns was generated for 16.5 years as shown in Table 6.1. This is *data 1*, which does not contain a trend and segments.

For the second hypothesis, one more procedure was added from the procedure for *data 1*. After applying the first and second steps, 50 (i.e., a trend) was added for the next year values as shown in Table 6.2 (e.g., the column of Jan. increases by 50). This is *data 2*, which contains a trend.

For the third hypothesis, after applying the first and second steps from the procedure for *data 1*, 100 (i.e., a trend) was added to the next year values. Then, eight consecutive monthly values starting from a random number were set to small numbers σ (i.e., segments) throughout the years as shown in Table 6.3 ($\sigma = 0$ in this test), which represents periods of no activity. This is *data 3*, which contains a trend and segments.

Table 6.1: Sample of *data 1*

Year	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
1	470	538	544	669	232	911	747	353	909	980	133	213
2	475	540	545	672	231	913	742	354	909	982	130	218
3	475	542	544	670	234	908	747	354	914	985	129	215
4	466	539	547	671	229	919	745	350	906	975	135	216
5	473	534	548	674	232	913	748	358	913	984	135	214
6	474	539	539	668	232	911	747	349	908	983	132	208
7	471	541	548	667	232	913	748	353	912	982	137	214
8	473	543	545	666	229	907	748	354	911	980	136	217
9	467	536	542	670	229	911	745	355	907	975	138	211
10	466	537	544	674	235	914	743	355	910	979	136	217
11	468	536	543	673	230	909	749	349	909	982	129	215
12	472	542	542	665	222	908	750	351	908	976	132	208
13	466	541	545	664	229	916	746	351	905	977	132	218
14	473	542	539	667	229	912	742	354	908	977	133	217
15	474	538	541	664	228	914	748	349	905	984	133	209
16	473	533	549	674	232	911	751	356	909	979	135	212
17	467	534	539	672	234	915						

For each data, a total of 20 data sets were generated and tested. The first 7 years of data were used as training data and the remaining 9.5 years of data were used as test data as shown in Figure 6.1.

6.4.2 Test results

The goal of this test is to construct a predictive model with the training data sets and predict future values (i.e., b_m in Equation (6.45) and (4.16)). The test data sets work as real values (i.e., d_m in Equation (6.45) and (4.16)). Table 6.4 shows the results of the data sets (*data 1, 2, 3*) in Section 4.1.

First, for *data 1* (data without a trend and segments), since the data sets are designed to be constant with some mild randomness, the constant rate method showed good prediction performance for the accuracy measure. The PUMLCA algorithm with both ETS (PUMLCA-ets) and ARIMA (PUMLCA-arima) also showed the similar level of accuracy and there is no significant difference between the constant rate method and PUMLCA (Mann-Whitney test, $\alpha = 0.05$, p-value=0.95). For the variance measure, since the constant rate method took the average rate for each month, monthly predictions of the constant rate method showed much lower accuracy than those of PUMLCA (Mann-Whitney test, $\alpha = 0.05$, p-value=0). This affects the prediction of the next quarter values (i.e., hypothesis 4) because lower monthly errors can give higher chances to predict specific periods with accuracy. For the next quarter values, the PUMLCA showed higher prediction accuracy (Mann-Whitney test, $\alpha = 0.05$, p-value=0). Therefore, the PUMLCA algorithm can provide accurate prediction capabilities for aggregated life cycle values (accuracy), seasonal values (variance) and values for specific periods with *data 1* in comparison to the constant rate method.

Second, for *data 2* (data with a trend), the constant rate method showed poor prediction performance in terms of the accuracy measure. On the other hand, the PUMLCA algorithm with both ETS and ARIMA showed good

Table 6.2: Sample of *data 2*

Year	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
1	975	872	965	976	799	449	681	169	399	728	614	725
2	1024	921	1010	1029	845	500	733	219	455	779	669	772
3	1077	973	1061	1070	893	549	786	271	502	828	713	823
4	1123	1022	1119	1129	940	605	832	312	549	872	765	871
5	1174	1077	1160	1179	991	659	885	365	600	928	813	923
6	1224	1117	1210	1224	1040	701	930	421	658	974	870	975
7	1273	1176	1268	1275	1095	751	978	462	698	1030	913	1025
8	1326	1220	1309	1325	1139	808	1029	522	751	1073	963	1079
9	1381	1271	1359	1379	1197	854	1078	567	805	1130	1011	1131
10	1427	1321	1419	1421	1248	899	1128	616	857	1179	1063	1181
11	1481	1367	1468	1472	1299	950	1180	671	900	1230	1117	1225
12	1526	1419	1515	1526	1340	1006	1229	712	953	1278	1162	1278
13	1575	1469	1561	1569	1393	1058	1278	769	1005	1328	1215	1328
14	1629	1527	1618	1625	1447	1099	1337	821	1056	1371	1268	1380
15	1677	1575	1667	1670	1495	1155	1378	872	1102	1420	1320	1423
16	1723	1623	1714	1722	1540	1209	1429	933	1153	1469	1372	1471
17	1770	1671	1760	1776	1592	1258						

prediction accuracy. There is no significant difference found between the real values and the results of PUMLCA-ets/arima (Mann-Whitney test, $\alpha = 0.05$, p-value=0.29/0.78). For the variance measure, monthly predictions of the constant rate method showed much lower accuracy than those of PUMLCA (Mann-Whitney test, $\alpha = 0.05$, p-value=0). This affects the prediction of the next quarter values. For the next quarter values, the PUMLCA showed higher prediction accuracy (Mann-Whitney test, $\alpha = 0.05$, p-value=0). Therefore, the PUMLCA algorithm can provide accurate prediction capabilities for aggregated life cycle values (accuracy), seasonal values (variance) and values for specific periods with *data 2* in comparison to the constant rate method.

Third, for *data 3* (data with a trend and segments), the constant rate method and the ETS method without the automatic segmentation algorithm (ets-no seg) showed poor prediction performance in terms of the accuracy measure. On the other hand, the ARIMA method without the automatic segmentation algorithm (arimai-no seg) and PUMLCA-ets/arima showed strong prediction accuracy. However, Table 6.5 zooms in their prediction performances using MAE, and it can be seen that the errors from the ARIMA method without the automatic segmentation algorithm were much higher than the those from the PUMLCA method. Due to the importance of the usage stage, the errors from the ARIMA method without the automatic segmentation are not acceptable, and this shows that the automatic segmentation algorithm can enhance the prediction result. Out of 20 samples, the PUMLCA-ets/arima showed the best performance. For the next quarter values, the PUMLCA method with the automatic segmentation algorithm showed higher prediction accuracy. Therefore, the proposed segmentation algorithm can improve the predictive model of PUMLCA with *data 3*.

Overall, the PUMLCA method with the automatic segmentation algorithm provided better prediction performance than the constant rate method for various data sets which are simulated from the observation of real data. This prediction improvement of usage modeling will help to estimate the environmental impact of the product of interest

Table 6.3: Sample of *data 3*

Year	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
1	σ	σ	σ	σ	σ	σ	155	129	643	313	σ	σ
2	σ	σ	σ	σ	σ	σ	257	233	746	409	σ	σ
3	σ	σ	σ	σ	σ	σ	355	333	848	518	σ	σ
4	σ	σ	σ	σ	σ	σ	452	429	944	610	σ	σ
5	σ	σ	σ	σ	σ	σ	558	525	1038	710	σ	σ
6	σ	σ	σ	σ	σ	σ	654	632	1141	813	σ	σ
7	σ	σ	σ	σ	σ	σ	752	734	1242	909	σ	σ
8	σ	σ	σ	σ	σ	σ	855	827	1344	1012	σ	σ
9	σ	σ	σ	σ	σ	σ	958	928	1445	1117	σ	σ
10	σ	σ	σ	σ	σ	σ	1053	1025	1542	1214	σ	σ
11	σ	σ	σ	σ	σ	σ	1160	1124	1643	1317	σ	σ
12	σ	σ	σ	σ	σ	σ	1253	1231	1743	1410	σ	σ
13	σ	σ	σ	σ	σ	σ	1354	1328	1839	1510	σ	σ
14	σ	σ	σ	σ	σ	σ	1450	1425	1943	1616	σ	σ
15	σ	σ	σ	σ	σ	σ	1553	1534	2044	1711	σ	σ
16	σ	σ	σ	σ	σ	σ	1656	1629	2143	1808	σ	σ
17	σ	σ	σ	σ	σ	σ						

Table 6.4: Test results

	Constant rate	ets-no seg	arima-no seg	PUMLCA-ets	PUMLCA-arima
data 1, average MAPE					
Accuracy	0.75			0.08	0.14
Variance	65.58			0.76	0.79
Next quarter value	13.84			0.25	0.24
data 2, average MAPE					
Accuracy	37.05			2.80	0.91
Variance	34.92			2.80	0.98
Next quarter value	22.06			0.74	0.29
data 3, average MAE					
Accuracy	30736	24462	1612	166	154
Variance	636	313	225	2	2
Next quarter value	1979	1017	139	10	9

more accurately. The example of the LCA with PUMLCA will be provided in the next section. The PUMLCA method could also provide prediction intervals while estimating a point forecast. For example, a point forecast of the next month is 1344 with the 80% prediction interval of [1330, 1359]. The prediction interval can show the uncertainty of time series usage models.

Table 6.5: MAEs over 20 data samples of *data 3*

	1	2	3	4	5	6	7	8	9	10
arima-no seg	1870	3005	855	1478	2295	2382	592	2200	965	156
PUMLCA-ets	58	1061	64	48	311	292	17	237	101	34
PUMLCA-arima	145	1044	16	24	293	224	59	173	102	66
	11	12	13	14	15	16	17	18	19	20
arima-no seg	558	865	1870	1464	829	2829	1170	2826	1971	2060
PUMLCA-ets	96	70	540	9	102	122	66	3	48	48
PUMLCA-arima	57	0	322	147	119	80	35	64	47	66

6.5 Illustrative Example: Agricultural Machinery Design

6.5.1 Background

In this section, the proposed algorithm, predictive usage mining for life cycle assessment (PUMLCA), is demonstrated with a case study of agricultural machines: current and new machine. The machines have more than 15,000 parts and weigh more than 20,000 kg. The current machine was updated to have a 10 % reduction of its environmental impact based on an improved fuel efficiency. This updated machine is called the new machine. The goal is to estimate the environmental impacts of the current and new machines in a real time horizon. Due to the data security issue, simulated data is used based on the observation of real data. Table 6.6 and 6.7 show simulated seven-year monthly data for fuel consumption and operating hours after preprocessing the raw sensor data.

Table 6.6: Monthly representation of fuel consumption (ℓ) data

Year	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
2007	9	0	0	0	0	0	0	2	600	3,400	5,000	250
2008	15	0	0	0	0	0	0	0	650	3,410	5,500	270
2009	17	0	0	0	0	0	0	0	660	3,450	5,550	280
2010	16	0	0	0	0	0	0	1	665	3,370	5,600	270
2011	14	0	0	0	0	0	0	1.5	660	3,430	5,650	275
2012	16	0	0	0	0	0	0	0	680	3,500	5,735	280
2013	17	0	0	0	0	0	0	2	700	3,570	5,800	285

In this case study, time series usage models from the historical sensor data will be utilized to calculate the environmental impacts for up to 10 to 20 years. Since the first stage (i.e., data preprocessing in Section 6.2.1) of PUMLCA is straightforward and simple, it was skipped in this section.

6.5.2 Seasonal Period Analysis

Instead of exploring all possible data representations (e.g., daily, weekly, quarterly, etc.), the focus was set on whether the simulated data showed a monthly seasonality. The periodogram was plotted using Equation (6.7) with the condition

Table 6.7: Monthly representation of operating hours (hr) data

Year	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
2007	1	0	0	0	0	0	0	0.2	35.2	100.6	152.3	15.1
2008	1.8	0	0	0	0	0	0	0	37.1	101.6	158.1	16.3
2009	2	0	0	0	0	0	0	0	38	105.3	159.3	17.8
2010	1.9	0	0	0	0	0	0	0.1	38.3	97.6	160.1	16.5
2011	1.7	0	0	0	0	0	0	0.2	38	103.5	162.2	17
2012	1.9	0	0	0	0	0	0	0	39	110.3	164.3	17.9
2013	2	0	0	0	0	0	0	0.22	41	115.2	165.2	18.2

of frequency greater than zero. Figure (6.7) shows that the maximum periodogram value can be achieved at the frequency of 0.0833 (i.e., period = 1/0.0833=12) for the fuel consumption data. Similarly, the operating hours data also indicate a period of 12.

6.5.3 Segmentation Analysis

The automatic segmentation algorithm (Algorithm 2) was applied to the two data sets in Table 6.6 and 6.7. As a penalty, the type I error of 0.05 was used for both data sets. First, for the fuel consumption data, a segment from February to August was identified as a shared segment since the same change points were detected (1, 8, 9, 10, 11, and 12 as seasonal time indexes) every year. Therefore, two segments were finally obtained, e.g., the shared segment (February$\sim$August) and the remaining segment (January, September$\sim$December). Second, for the operating hours data, the segment from January to August was identified as a shared segment. The same change points were detected (8, 9, 10, 11, and 12 as seasonal time indexes) every year. Therefore, two segments were finally obtained.

6.5.4 Time Series Analysis

The automatic forecasting algorithm in Section 6.2.4 was applied to the original data sets (i.e., without segmentation) and the results of the automatic segmentation in Section 6.5.3. Table 6.8 shows the results. For example, the original fuel consumption data is fitted as a seasonal AR model with a seasonal differencing and a drift using ARIMA. The first segment data (segment 1) shows a combination of seasonal AR and MA models without a drift. The second segment data (segment 2) shows only a seasonal differencing operation with a drift. The original fuel consumption data is also fitted as an additive error and seasonal component model using ETS. The first segment data shows an additive error and seasonal component model again. The second segment data shows an additive trend, multiplicative error and seasonal component model.

Table 6.9 shows the comparison of forecasts after 10 years (i.e., 2024) for fuel consumption data using the fitted time series models in Table 6.8. The second column represents the usage of the automatic segmentation algorithm.

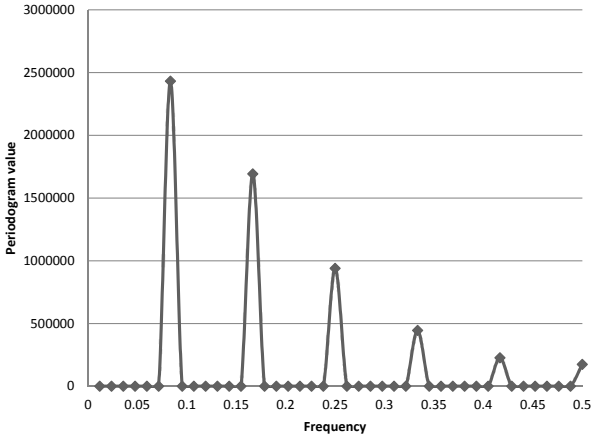


Figure 6.7: Periodogram for fuel consumption

It can be observed that the automatic segmentation algorithm can distinguish low activity periods and help to capture patterns more clearly.

6.5.5 Predictive LCA

LCA for Current Machine

The PUMLCA-ets models (with two segments) of fuel consumption, N_{ft}, and operating hours, OH_t, in Table 6.8 were used as usage models of the agricultural machine. For predictive LCA, starting from January 2014, forecasts were built up to December 2024 (i.e., 10 years) and up to December 2034 (i.e., 20 years). For environmental impact calculation, Eco-Indicator 99 method (EI-99) [155] was used, which is one of widely used methods in LCA and provides a single score (Point) from pre-defined damage categories such as human health, ecosystem quality, and resource.

In the manufacturing stage, the environmental impact was assumed as 12,000 Pt. In the usage stage, the density of diesel fuel was assumed as 0.85 kg/liter and emission rates was given in Table 6.10. The idling and nonidling ratio (20%/80%) was calculated using averages of seven-year operating hours by work modes. In the maintenance stage, the assumptions on the replacement cycle of major parts and minor parts are as follows [1]: tires (3,000 hours), transmission (3,000 hours), hydraulic components (3,000 hours), engine (5,000 hours), axles (5,000 hours), and minor parts such as oils, greases, filters, etc. (specified cycle). In the end-of-life stage, the following assumptions were made: steel (90% recycle and 10% landfill), iron (90% recycle and 10% landfill), and others (80% landfill and 20% incineration).

Based on Equation (6.35), (6.36), (6.37) and (6.38), a predictive LCA result of the current machine in the real time horizon (January 2014$\sim$December 2034) was estimated as shown in Figure 6.8. The impact of the manufacturing

Table 6.8: Results of time series analysis

		ARIMA	ETS
Fuel consumption data	Original	$(1-0.41B^{12})(1-B^{12})y_t = 1.53 + e_t$	$y_t = l_{t-1} + s_{t-12}$ $l_t = l_{t-1} + 0.06\varepsilon_t$ $s_t = s_{t-12} + 10^{-4}\varepsilon_t$
	Segment 1 (Feb.∼Aug.)	$(1+0.28B^{7})(1-B^{7})y_t = (1-0.28B^{4})e_t$	$y_t = l_{t-1} + s_{t-7}$ $l_t = l_{t-1} + 0.001\varepsilon_t$ $s_t = s_{t-7} + 2 \cdot 10^{-4}\varepsilon_t$
	Segment 2 (Jan., Sept.∼Dec.)	$(1-B^{5})y_t = 7.42 + e_t$	$y_t = (l_{t-1} + b_{t-1})s_{t-5}$ $l_t = (l_{t-1} + b_{t-1})(1+0.395\varepsilon_t)$ $b_t = b_{t-1} + 0.098(l_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-5}(1+10^{-4}\varepsilon_t)$
Operating hours data	Original	$(1-B^{12})y_t = (1+0.21B)e_t$	$y_t = l_{t-1} + s_{t-12}$ $l_t = l_{t-1} + 0.29\varepsilon_t$ $s_t = s_{t-12} + 3 \cdot 10^{-4}\varepsilon_t$
	Segment 1 (Jan.∼Aug.)	$(1-B^{8})y_t = (1-0.67B)(1-0.64B^{8})e_t$	$y_t = l_{t-1} + s_{t-8}$ $l_t = l_{t-1} + 10^{-4}\varepsilon_t$ $s_t = s_{t-8} + 0.03\varepsilon_t$
	Segment 2 (Sept.∼Dec.)	$(1-B^{4})y_t = 0.38 + e_t$	$y_t = l_{t-1} + s_{t-4}$ $l_t = l_{t-1} + 0.12(l_{t-1} + s_{t-4})\varepsilon_t$ $s_t = s_{t-4} + 0.88(l_{t-1} + s_{t-4})\varepsilon_t$

Table 6.9: Comparison of forecasts after 10 years for fuel consumption (ℓ) data

Method	Segmentation	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
ARIMA	No	189	171	171	171	171	171	171	175	885	3,790	6,016	460
PUMLCA-arima	Yes	388	0	0	0	0	0	0	0.9	1,071	3,941	6,171	656
ETS	No	45	41.4	36.4	40	34	29	50	41	695	3,473	5,564	310
PUMLCA-ets	Yes	18.4	0	0	0	0	0	0	0.9	814	4,240	6,649	328

stage was the same regardless of time horizons since it is a one-time event. On the other hand, the impacts of the usage, maintenance, and end-of-life stage were varied by time. Similar to previous LCA studies, the impact of the usage stage accounted for the majority of the environmental impact. The impact of the maintenance stage showed a big increase since major parts (engine and axles) were replaced after 10 years. It should be noted that the two usage models (PUMLCA and constant rate method) were used for the usage stage in order to show the impact of prediction accuracy in Section 6.4 (PUMLCA was also used for the maintenance and end-of-life stages). The data in this case study was similar to the third hypothesis in Section 6.4 (i.e., data with increasing trend and segments) so that it can be expected that the constant rate method would underestimate the impact (about 17,000 Pt over 20 years), which is greater than the impact of the manufacturing stage. If the data is quite constant, a similar result between PUMLCA and the constant rate method would be produced as seen in Section 6.4 (i.e., data without trend and segments, aggregated life cycle values). Furthermore, the top of Figure 6.8 shows the 80% prediction intervals of the usage impact by PUMLCA. Unlike the constant rate method, PUMLCA can provide the uncertainty of its predictive model.

Table 6.10: Assumptions on emission rates (g/hr) [1]

Type	Nonidling (80%)	Idling (20%)	Average
Nitrogen oxides (NOx)	372.73	143.16	326.82
Particulate matter (PM)	1.76	0.67	1.54
Carbon monoxide (CO)	23.84	9.16	20.9
Hydrocarbons (HC)	5.42	2.08	4.75
Sulfur dioxide (SO_2)	0.99	0.43	0.89
Carbon dioxide (CO_2)	150829.6	065427.83	133749.3

LCA for New Machine

New machines were assumed to be designed based on the current machines with the target of 10% reduction of environmental impact over 20 years. It needs to utilize the usage data of the current machines with the improvement ratio, δ_{N_f} and δ_{OH} as shown in Figure 6.6. Similar to the current machine, predictive LCA was conducted starting from January 2014 up to December 2024 (i.e., 10 years) and up to December 2034 (i.e., 20 years) with the EI-99 method.

In the manufacturing stage, the environmental impact was assumed to be increased to 14,500 Pt (20.8%) due to the additional power sources. The other assumptions of the usage, maintenance and end-of-life stage were similar to the current machine. The unit of work was the square meter (m^2) and the performance test was conducted to compare the new machine and the current machine. The improvement ratio for fuel consumption δ_{N_f} was 0.8 and the improvement ratio for operating hours δ_{OH} was 0.85.

Based on Equation (6.41), (6.42), (6.43) and (6.44), the predictive LCA result of the new machine in the real time horizon (January 2014$\sim$December 2034) was estimated as shown in Figure 6.9.

Table 6.11 shows the comparison of the two LCA results of the current and new machine. Although the impact from the manufacturing stage was increased (20.8%) for the new machine, the total impact was reduced mainly from the usage stage. It should be noted that the result depends on the lifespan of machines. 8.4% of environmental impact reduction was expected for 10 years and 11.3% for 20 years, which satisfies the target of 10% reduction of environmental impact over 20 years. Sensitivity analysis can be applied to find the minimum values of the improvement ratio, δ_{N_f} and δ_{OH} to satisfy the target. In conclusion, the proposed algorithm, PUMLCA, captured usage patterns from large-scale sensor data with the automatic segmentation algorithm and time series analysis, and could assess environmental impact of a complex system in a real time horizon.

Table 6.11: Comparison of current and new machines (EI-99, Pt)

	Manufacturing		Usage		Maintenance		End-of-life		Total	
	10 yr.	20 yr.	10 yr.	20 yr.	10 yr.	20 yr.	10 yr.	20 yr.	10 yr.	20 yr.
Current machine	12,000	12,000	41,706	84,002	9,295	22,890	476	805	63,477	119,697
New machine	14,500	14,500	33,763	67,961	9,400	22,900	480	820	58,143	106,181

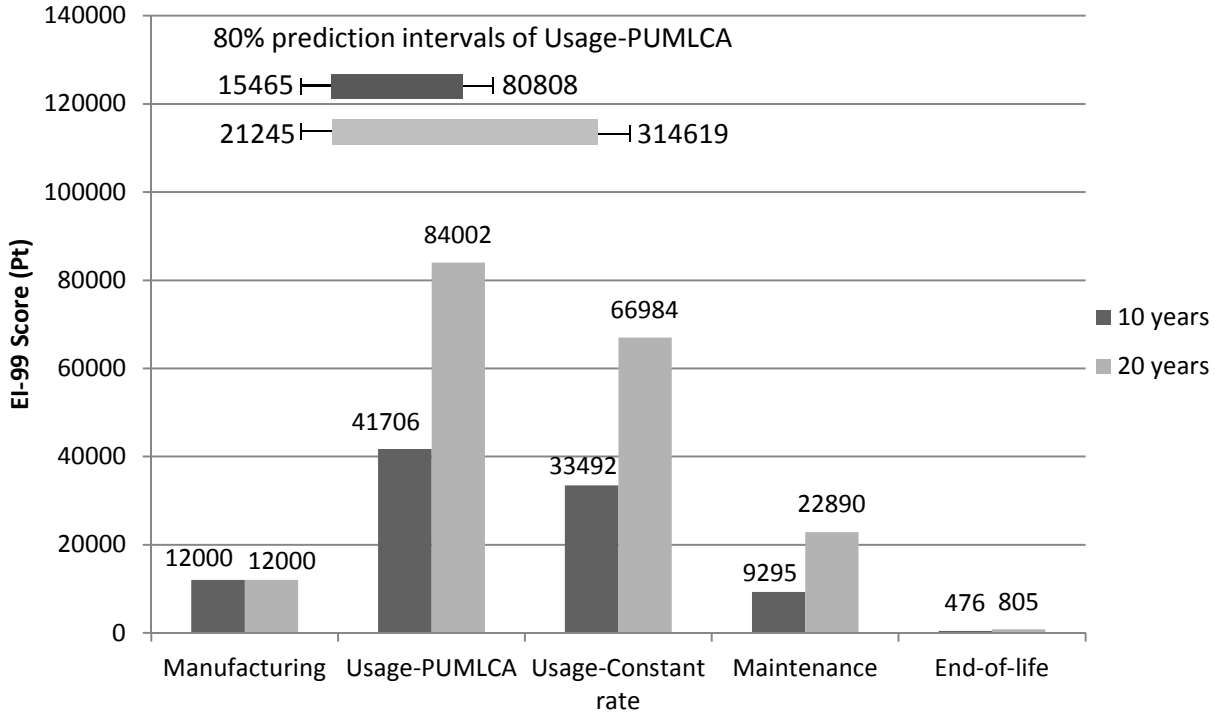


Figure 6.8: Predictive LCA results for current machine

6.6 Conclusion

In this chapter, the predictive usage mining for life cycle assessment (PUMLCA) algorithm is proposed to model the usage stage for the LCA of products. By defining usage patterns as trend, seasonality, and level from a time series of usage information, predictive LCA can be conducted in a real time horizon, which can provide more accurate results of LCA. Large-scale sensor data of product operation was analyzed to mine usage patterns and build a usage model for LCA. The PUMLCA algorithm includes handling missing and abnormal values, seasonal period analysis, segmentation analysis, time series analysis, and predictive LCA. In order to mine important usage patterns more effectively from a time series, the automatic segmentation algorithm is developed based on change point analysis.

The prediction performance test results with various data sets showed that the predictive model from the PUMLCA method can provide better prediction accuracy than the constant rate method. The automatic segmentation algorithm magnified important patterns and helped to predict future values more accurately.

Two different design problems were formulated to incorporate the usage model from the PUMLCA method in predictive LCA. The case study of agricultural machinery showed how to apply the PUMLCA method for the predictive LCA of complex systems. The environment impacts of both current machines and new machines could be estimated and compared.

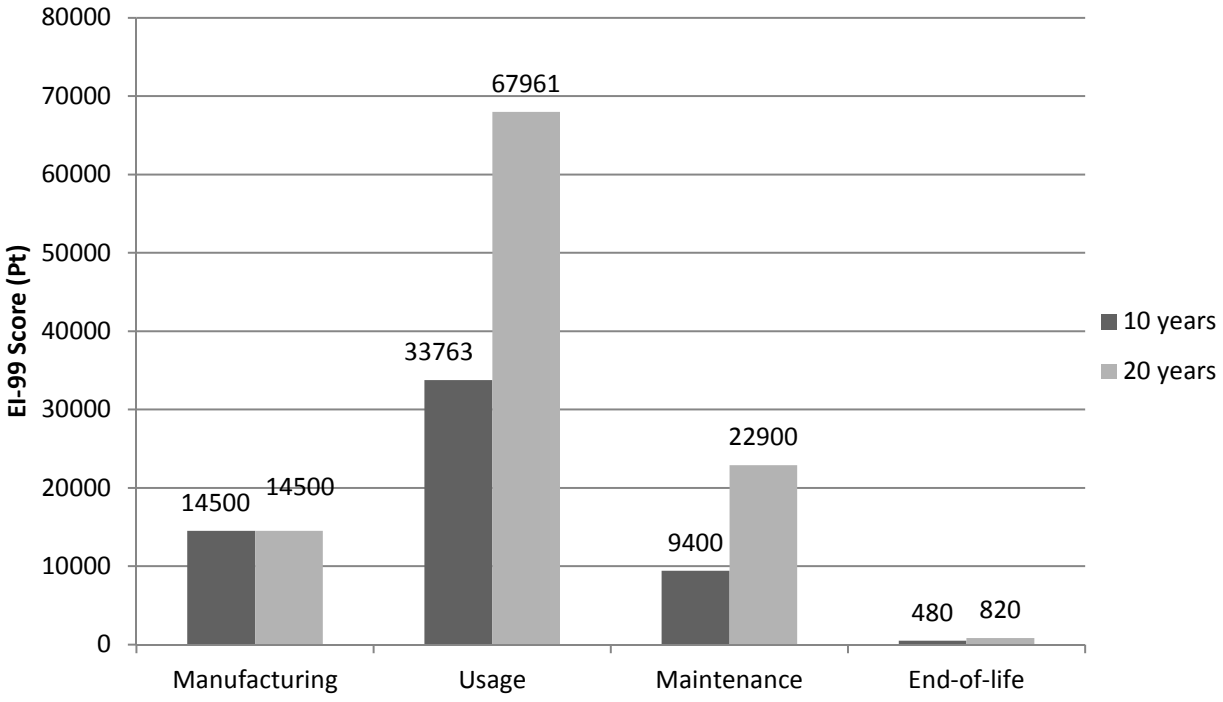


Figure 6.9: Predictive LCA results for new machine

In the future, various data sets from different products can be tested with the PUMLCA algorithm. The current model, which considers only a single type of machinery, can be extended to multiple types of machinery. In order to perform LCA with multiple types of machinery, hierarchical time series modeling and forecasting may be helpful [156].

The next chapter will briefly summarize all the chapters and discuss future research directions.

Chapter 7

Closure

7.1 Summary

This dissertation discusses data analytics methods for better system design while considering the challenges of large-scale data characterized by the four dimensions (volume, variety, velocity and veracity). Though the four dimensions of large-scale data in the domain of system design cause distinctive challenges to design engineers, they also represent new opportunities to improve various design decisions.

Predictive design analytics is proposed not only as a paradigm to motivate design engineers in the era of Big Data but also as a framework to be applied in design problems. Economical life cycle design (Chapters 3 and 4) requires the modeling of time-varying customer preferences. The demand trend mining (DTM) and continuous preference trend mining (CPTM) algorithms are developed to serve this need. Product family design (Chapter 5) demands the combination of data-driven and market-driven approaches to determine product family architectures. The predictive, data-driven family design (PDPFD) algorithm is developed to find the optimal architecture design in the near future. Sustainable design (Chapter 6) necessitates the modeling of product usage and forecasting. The predictive usage mining for life cycle assessment (PUMLCA) algorithm is developed to provide a new usage model from sensor data, which can estimate future environmental impacts of current and new machines. The benefits and effectiveness of these predictive design analytics methods are demonstrated with electronic products, mechanical parts and complex agricultural machines.

In addition to the contributions of the proposed models as design support systems, Figure 7.1 emphasizes the contributions of them as data analytics. The proposed models are either extended from existing models or newly developed. Furthermore, they provides good properties that other analytics models could not provide as shown in Figure 7.1.

Though the proposed predictive design analytics in engineering system design shows some potential and opportunities, there are more research areas to be explored. Apart from the design problems that are discussed in this dissertation, there are more interesting and challenging design problems that are yet to be explored. Tackling new design problems and developing necessary data analytics methods would be the immediate future work of this dissertation.

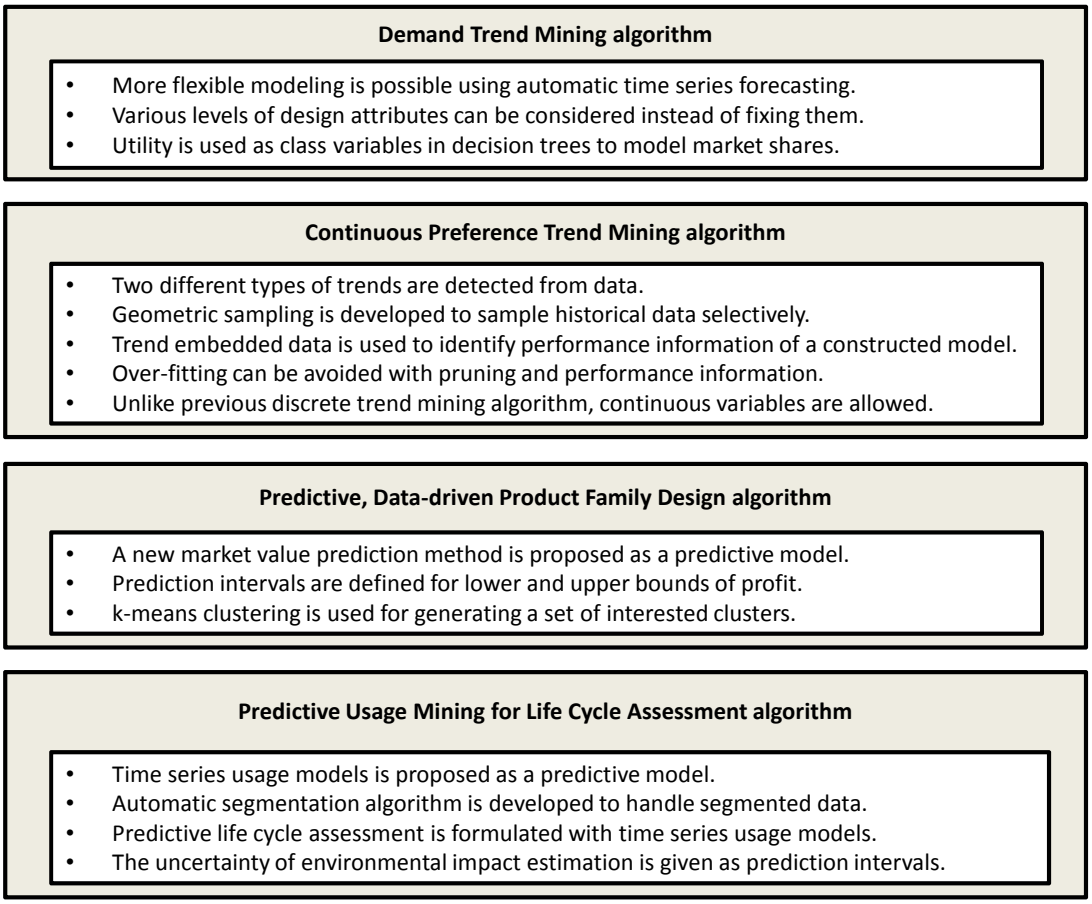


Figure 7.1: Contributions of data analytics methods in this dissertation

In addition to this, in the next section, two future research directions are suggested.

7.2 Future Work

7.2.1 Predictive Modeling

More theoretical foundations of predictive modeling can lead to the enhancement of predictive design analytics methods. Shmueli [157] discussed predictive modeling in comparison with explanatory modeling. The critical question is how to quantify predictive power and predictive accuracy. Furthermore, multicollinearity should be properly addressed. The possible application would be predicting the quantity, quality and timing of product returns in economical life cycle design [1]. Note that this dissertation assumes special market environments such as take-back programs in Chapter 3 or lease contracts in Chapter 4 instead of handling the issue with statistical models.

[1] The related work will be presented in [158].

7.2.2 Big Data Analytics

Another future research direction is Big Data analytics. In this dissertation, only data volumes that one computer can handle (but much larger than traditional data from a controlled environment) are utilized under the assumption that proper preprocessing can be applied, and the data is related to the domain of engineering design. If the volumes exceed the capability of one computer, parallel, distributed system can be built using the Hadoop framework. The Hadoop framework provides modules such as distributed storage (Hadoop Distributed File System), parallel processing (Hadoop MapReduce) and data mining library (Mahout). Furthermore, the efficiency of data analytics methods is a challenge in Big Data. Data reduction techniques such as support vectors in support vector machines and principle components in principle components analysis can be utilized to reduce the necessity of processing all data.

Chapter 8

References

[1] M. Kwak and H.M. Kim. Economic and environmental impacts of product service lifetime: A life-cycle perspective. In Horst Meier, editor, *Product-Service Integration for Sustainable Solutions*, Lecture Notes in Production Engineering, pages 177–189. Springer Berlin Heidelberg, 2013.

[2] B.V. Pre Consultant. The eco-indicator 99 manual for designers: A damage oriented method for life cycle impact assessment. Technical report, Ministry of Housing, Spatial Planning and the Environment, Den Haag, The Netherlands, 2000. http://www.pre-sustainability.com/download/manuals/EI99_Manual.pdf.

[3] T.W. Simpson, J.R. Maier, and F. Mistree. Product platform design: method and application. *Research in Engineering Design*, 13(1):2–22, 2001.

[4] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–21, 2004.

[5] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. 2013. Accessed: Jan. 2014.

[6] M. Schroeck, R. Shockley, J. Smart, D. Romero-Morales, and P. Tufano. Analytics: The real-world use of big data. Ibm institute for business value - executive report, IBM Institute for Business Value, 2012.

[7] E. Siegel and T.H. Davenport. *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*. EBL ebooks online. Wiley, 2013.

[8] W.W. Eckerson. Predictive analytics: Extending the value of your data warehousing investment. Tdwi best practices report, TDWI, 2007. http://www.sas.com/events/cm/174390/assets/102892_0107.pdf.

[9] T.W. Miller. *Modeling Techniques in Predictive Analytics: Business Problems and Solutions with R*. Pearson Education, 2013.

[10] M. Böttcher, M. Spott, and R. Kruse. Predicting future decision trees from evolving data. In *Proceedings of ICDM '08*, pages 33–42, 2008.

[11] M. Böttcher. Contrast and change mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):215–230, 2011.

[12] R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intell. Data Anal.*, 8(3):281–300, August 2004.

[13] K. McGarry. A survey of interestingness measures for knowledge discovery. *Knowledge Eng. Review*, 20(1):39–61, 2005.

[14] C.S. Tucker and H.M. Kim. Trend mining for predictive product design. *Journal of Mechanical Design*, 133(11):111008, 2011.

[15] V.N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[16] Environmental Protection Agency. Electronics waste management in the united states through 2009. Report EPA 530-R-11-002, U.S. EPA, May 2011.

[17] M.S. Sodhi and B. Reimer. Models for recycling electronics end-of-life products. *OR Spektrum*, 23(1):97–115, 2001.

[18] B.K. Fishbein. EPR: What does it mean? Where is it headed? *P2: Pollution Prevention Review*, 8(4):43–55, 1998.

[19] Product Stewardship Institute. Extended producer responsibility state laws. http://productstewardship.us (Accessed: May 2013), 2012.

[20] S.A. Wagner. *Understanding Green Consumer Behaviour: A Qualitative Cognitive Approach*. Consumer Research and Policy Series. Taylor & Francis Group, 2003.

[21] Environmental Protection Agency. Benefits of the remanufacturing exclusion: Background document in support of the definition of solid waste rule, June 2011.

[22] M. Hucal. Product recycling creates multiple lives for caterpillar machines. *Peoria Magazines*, September 2008.

[23] A. King, J. Miemczyk, and D. Bufton. Photocopier remanufacturing at xerox uk a description of the process and consideration of future policy issues. In Daniel Brissaud, Serge Tichkiewitch, and Peggy Zwolinski, editors, *Innovation in Life Cycle Engineering and Sustainable Development*, pages 173–186. Springer Netherlands, 2006.

[24] D. Parker and P. Butler. An introduction to remanufacturing, 2007. http://www.remanufacturing.org.uk (Accessed: May 2013).

[25] S.K. Fixson. Assessing product architecture costing: Product life cycles, allocation rules, and cost models. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2004)*, number DETC2004-57458, Salt Lake City, USA, 2004.

[26] P. Duverlie and J.M. Castelain. Cost estimation during design step: Parametric method versus case based reasoning method. *The International Journal of Advanced Manufacturing Technology*, 15(12):895–906, 1999.

[27] K.K. Seo, J.H. Park, D.S. Jang, and D. Wallace. Approximate estimation of the product life cycle cost using artificial neural networks in conceptual design. *The International Journal of Advanced Manufacturing Technology*, 19(6):461–471, 2002.

[28] M.S. Hundal. *Mechanical Life Cycle Handbook: Good Environmental Design and Manufacturing*. Mechanical engineering. Marcel Dekker, 2001.

[29] M. Kwak. *GREEN PROFIT DESIGN FOR LIFECYCLE*. PhD thesis, University of Illinois at Urbana-Champaign, June 2012.

[30] S.W. Lye, S.G. Lee, and M.K. Khoo. A design methodology for the strategic assessment of a product's eco-efficiency. *International Journal of Production Research*, 39(11):2453–2474, 2001.

[31] M. O'Shea. Design for environment in conceptual product design a decision model to reflect environmental issues of all life-cycle phases. *The Journal of Sustainable Product Design*, 2(1):11–28, 2002.

[32] R. Holt and C. Barnes. Towards an integrated approach to design for x: an agenda for decision-based dfx research. *Research in Engineering Design*, 21:123–136, 2010. 10.1007/s00163-009-0081-6.

[33] C.M. Rose, A. Stevels, and K. Ishii. A new approach to end-of-life design advisor (ELDA). In *Electronics and the Environment, 2000. ISEE 2000. Proceedings of the 2000 IEEE International Symposium on*, pages 99–104, 2000.

[34] C.M. Rose, K. Ishii, and A. Stevels. Influencing design to improve product end-of-life stage. *Research in Engineering Design*, 13:83–93, 2002.

[35] D. Mangun and D.L. Thurston. Incorporating component reuse, remanufacture, and recycle into product portfolio design. *Engineering Management, IEEE Transactions on*, 49(4):479–490, nov 2002.

[36] M. Kwak and H.M. Kim. Evaluating end-of-life recovery profit by a simultaneous consideration of product design and recovery network design. *Journal of Mechanical Design*, 132(7):071001, 2010.

[37] M. Kwak and H.M. Kim. Assessing product family design from an end-of-life perspective. *Engineering Optimization*, 43(3):233–255, 2011.

[38] Y. Zhao and D. Thurston. Integrating end-of-life and initial profit considerations in product life cycle design. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2010)*, Montreal, Quebec, Canada, 2010. DETC2010-28830.

[39] M. Kwak and H.M. Kim. Design for life-cycle profit with simultaneous consideration of initial manufacturing and end-of-life remanufacturing. *Engineering Optimization*, 0(0):1–18, 0.

[40] T.W. Simpson, J. Jiao, Z. Siddique, and K. Hölttä-Otto. *Advances in Product Family and Product Platform Design: Methods & Applications*. Springer New York, 2014.

[41] T.W. Simpson, A. Bobuk, L.A. Slingerland, S. Brennan, D. Logan, and K. Reichard. From user requirements to commonality specifications: an integrated approach to product family design. *Research in Engineering Design*, 23(2):141–153, 2012.

[42] O.L. de Weck, E. Suh, and D.D. Chang. Product family and platform portfolio optimization. In *2003 ASME Design Engineering Technical Conference*, number DETC03/DAC-48721, Chicago, Illinois, September 2-6 2003. American Society of Mechanical Engineers.

[43] K.T. Ulrich and S.D. Eppinger. *Product design and development*. McGraw-Hill, 2012.

[44] Z. Pirmoradi, Wang G.G., G. Gary Wang, and T.W. Simpson. A review of recent literature in product family design and platform-based product development. In T.W. Simpson, J. Jiao, Z. Siddique, and K. Hölttä-Otto, editors, *Advances in Product Family and Product Platform Design*, pages 1–46. Springer New York, 2014.

[45] T.W. Simpson. Product platform design and customization: Status and promise. *AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18:3–20, 2 2004.

[46] Z. Dai and M.J. Scott. Product platform design through sensitivity analysis and cluster analysis. *Journal of Intelligent Manufacturing*, 18(1):97–113, 2007.

[47] C. Chen and L. Wang. Product platform design through clustering analysis and information theoretical approach. *International Journal of Production Research*, 46(15):4259–4284, 2008.

[48] R.U. Nayak, W. Chen, and T.W. Simpson. A variation-based method for product family design. *Engineering Optimization*, 34(1):65–81, 2002.

[49] A. Messac, M.P. Martinez, and T.W. Simpson. Introduction of a Product Family Penalty Function Using Physical Programming. *Journal of Mechanical Design*, 124(2):164–172, 2002.

[50] D.A. Collier. The measurement and operating benefits of component part commonality. *Decision Sciences*, 12(1):85–96, 1981.

[51] J.G. Wacker and M. Treleven. Component part standardization: An analysis of commonality sources and indices. *Journal of Operations Management*, 6(2):219–244, 1986.

[52] S. Kota, K. Sethuraman, and R. Miller. A metric for evaluating design commonality in product families. *Journal of Mechanical Design*, 122(4):403–410, 1998.

[53] H.J. Thevenot and T.W. Simpson. A comprehensive metric for evaluating component commonality in a product family. *Journal of Engineering Design*, 18(6):577–598, 2007.

[54] J.P. Gonzalez-Zugasti, K.N. Otto, and J.D. Baker. A method for architecting product platforms. *Research in Engineering Design*, 12(2):61–72, 2000.

[55] B. D'Souza and T.W. Simpson. A genetic algorithm based method for product family design optimization. *Engineering Optimization*, 35(1):1–18, 2003.

[56] A. Khajavirad and J.J. Michalek. A decomposed gradient-based approach for generalized platform selection and variant design in product family optimization. *Journal of Mechanical Design*, 130(7):1–8, 2008.

[57] A. Khajavirad, J.J. Michalek, and T.W. Simpson. An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality. *Structural and Multidisciplinary Optimization*, 39(2):187–201, 2009.

[58] K. Train. *Discrete choice methods with simulation*. Discrete Choice Methods with Simulation. Cambridge University Press, 2003.

[59] H.J. Wassenaar and W. Chen. An approach to decision-based design with discrete choice analysis for demand modeling. *Journal of Mechanical Design*, 125(3):490–497, 2003.

[60] H.J. Wassenaar, W. Chen, J. Cheng, and A. Sudjianto. Enhancing discrete choice demand modeling for decision-based design. *Journal of Mechanical Design*, 127(4):514–523, 2005.

[61] D. Kumar, W. Chen, and T.W. Simpson. A market-driven approach to product family design. *International Journal of Production Research*, 47(1):71–104, 2009.

[62] B. Agard and A. Kusiak. Data-mining-based methodology for the design of product families. *International Journal of Production Research*, 42(15):2955–2969, 2004.

[63] S.K. Moon, S R.T. Kumara, and T.W. Simpson. Data mining and fuzzy clustering to support product family design. In *2006 ASME Design Engineering Technical Conference*, number DETC2006-99287, Philadelphia, Pennsylvania, September 10-13 2006. American Society of Mechanical Engineers.

[64] C.S. Tucker, H.M. Kim, D.E. Barker, and Y. Zhang. A relieff attribute weighting and X-means clustering methodology for top-down product family optimization. *Engineering Optimization*, 42(7):593–616, 2010.

[65] K.Y. Chan, C.K. Kwong, and B.Q. Hu. Market segmentation and ideal point identification for new product design using fuzzy data compression and fuzzy clustering methods. *Applied Soft Computing*, 12(4):1371–1378, 2012.

[66] G. Rebitzer, T. Ekvall, R. Frischknecht, D. Hunkeler, G. Norris, T. Rydberg, W.-P. Schmidt, S. Suh, B.P. Weidema, and D.W. Pennington. Life cycle assessment: Part 1: Framework, goal and scope definition, inventory analysis, and applications. *Environment International*, 30(5):701–720, 2004.

[67] G. Finnveden, M.Z. Hauschild, T. Ekvall, J. Guine, R. Heijungs, S. Hellweg, A. Koehler, D. Pennington, and S. Suh. Recent developments in life cycle assessment. *Journal of Environmental Management*, 91(1):1–21, 2009.

[68] J.B. Guinée. *Handbook on Life Cycle Assessment: Operational Guide to the ISO Standards*. Eco-Efficiency in Industry and Science. Springer, 2002.

[69] John Reap, Felipe Roman, Scott Duncan, and Bert Bras. A survey of unresolved problems in life cycle assessment. part 1: Goal and scope and inventory analysis. *The International Journal of Life Cycle Assessment*, 13(4):290–300, 2008.

[70] John Reap, Felipe Roman, Scott Duncan, and Bert Bras. A survey of unresolved problems in life cycle assessment. part 2: Impact assessment and interpretation. *The International Journal of Life Cycle Assessment*, 13(5):374–388, 2008.

[71] A. Levasseur, P. Lesage, M. Margni, L. Deschłnes, and R. Samson. Considering time in lca: Dynamic lca and its application to global warming impact assessments. *Environmental Science & Technology*, 44(8):3169–3174, 2010.

[72] R. Memary, D. Giurco, G. Mudd, and L. Mason. Life cycle assessment: a time-series analysis of copper. *Journal of Cleaner Production*, 33(0):97–108, 2012.

[73] P. Collet, L. Lardon, J. Steyer, and A. Hlias. How to take time into account in the inventory step: a selective introduction based on sensitivity analysis. *The International Journal of Life Cycle Assessment*, 19(2):320–330, 2014.

[74] J. L. Sullivan and E. Cobas-Flores. Full vehicle lcas: a review. In *Proceedings of the 2001 Environmental Sustainability Conference and Exhibition*, pages 99–114, Graz, Austria, 2001.

[75] M. Kwak, L. Kim, O. Sarvana, H.M. Kim, P. Finamore, and H. Hazewinkel. Life cycle assessment of complex heavy duty equipment. In *ASME International Symposium on Flexible Automation (ISFA2012)*, number ISFA2012-7180, St. Louis, USA, 2012.

[76] C. Telenko and C. Seepersad. Probabilistic graphical models as tools for evaluating the impact of usage-context on the environmental performance of products. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2012)*, number DETC2012-71160, Chicago, USA, 2012.

[77] J. Lee, H. Cho, B. Choi, J. Sung, S. Lee, and M. Shin. Life cycle assessment of tractors. *The International Journal of Life Cycle Assessment*, 5(4):205–208, 2000.

[78] B. Choi, H. Shin, S. Lee, and T. Hur. Life cycle assessment of a personal computer and its effective recycling rate (7 pp). *The International Journal of Life Cycle Assessment*, 11(2):122–128, 2006.

[79] T. Li, Z. Liu, H. Zhang, and Q. Jiang. Environmental emissions and energy consumptions assessment of a diesel engine from the life cycle perspective. *Journal of Cleaner Production*, 53(0):7–12, 2013.

[80] J. Ma, M. Kwak, and H.M. Kim. Pre-life and end-of-life combined profit optimization with predictive product lifecycle design. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2012)*, number DETC2012-70528, Chicago, USA, 2012.

[81] J. Ma, M. Kwak, and H.M. Kim. Demand trend mining for predictive life cycle design. *Journal of Cleaner Production*, 68(0):189–199, 2014.

[82] A. Labrinidis and H.V. Jagadish. Challenges and opportunities with big data. *Proc. VLDB Endow.*, 5(12):2032–2033, August 2012.

[83] C.S. Tucker and H.M. Kim. Optimal product portfolio formulation by merging predictive data mining with multilevel optimization. *Journal of Mechanical Design*, 130(4):991–1000, 2008.

[84] D. Van Horn, A. Olewnik, and K. Lewis. Design analytics: Capturing, understanding and meeting customer needs using big data. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2011)*, number DETC2012-71038, 2012.

[85] J.A. PEARCE. In with the old. Technical report, Wall Street Journal, October 2008.

[86] R.T. Lund. *Remanufacturing : the experience of the United States and implications for developing countries*. World Bank, Washington, D.C., U.S.A., 1984.

[87] M. Charter and C. Gray. Remanufacturing and product design: designing for the 7th generation. Report, Centre for Sustainable Design, May 2007.

[88] P.J. Newcomb, B. Bras, and D.W. Rosen. Implications of modularity on product design for the life cycle. *Journal of Mechanical Design*, 120(3):483–490, 1998.

[89] Cellular-Recycler. Sustainability within the used cellular phone industry. Report, Cellular Recycler, January 2011. http://www.cellularrecycler.com/wp/wp-content/uploads/2011/01/CR-Sustainability-Report.pdf.

[90] R. Deng, E. Williams, and C. Babbitt. Hybrid life cycle assessment of energy use in laptop computer manufacturing. In *Sustainable Systems and Technology, 2009. ISSST '09. IEEE International Symposium on*, page 1, may 2009.

[91] W.L. Moore, J.J. Louviere, and R. Verma. Using conjoint analysis to help design product platforms. *Journal of Product Innovation Management*, 16(1):27–39, 1999.

[92] M.D. Grissom, A.D. Belegundu, A. Rangaswamy, and G.H. Koopmann. Conjoint-analysis-based multiattribute optimization: application in acoustical design. *Structural and Multidisciplinary Optimization*, 31:8–16, 2006.

[93] C.S. Tucker and H.M. Kim. Data-Driven Decision Tree Classification for Product Portfolio Design Optimization. *Journal of Computing and Information Science in Engineering*, 9, 2009.

[94] L.O. Hall, N. Chawla, and K.W. Bowyer. Decision tree learning on very large data sets. In *In IEEE Conference on Systems, Man and Cybernetics*, pages 2579–2584, 1998.

[95] Z. Yu, F. Haghighat, B.C.M. Fung, and H. Yoshino. A decision tree method for building energy demand modeling. *Energy and Buildings*, 42(10):1637–1646, 2010.

[96] R. Hyndman, A.B. Koehler, J.K. Ord, and R.D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer-Verlag Berlin Heidelberg, 2008.

[97] R.J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 2008.

[98] C.S. Tucker, C. Hoyle, H.M. Kim, and W. Chen. A comparative study of data-intensive demand modeling techniques in relation to product design and development. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2009)*, number DETC2009-87049, San Diego, CA, USA, 2009.

[99] M. Kwak and H.M. Kim. Market-driven positioning of remanufactured product for design for remanufacturing with part upgrade. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2011)*, number DETC2011-48432, 2011.

[100] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers, 1993.

[101] E. Harris. Information gain versus gain ratio: A study of split method biases. In *ISAIM*, 2002.

[102] G.E.P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1976.

[103] T.H. Naylor, T.G. Seaks, and D.W. Wichern. Box-jenkins methods: An alternative to econometric models. *International Statistical Review / Revue Internationale de Statistique*, 40(2):pp. 123–137, 1972.

[104] J.G. De Gooijer and R.J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006.

[105] M.D. Geurts and I.B. Ibrahim. Comparing the box-jenkins approach with the exponentially smoothed forecasting model application to hawaii tourists. *Journal of Marketing Research*, 12(2):pp. 182–188, 1975.

[106] R.J. Hyndmana, A.B. Koehlerb, R.D. Snydera, and S. Grosea. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002.

[107] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[108] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[109] USGS. Recycled cell phones-a treasure trove of valuable metals. Fact sheet 2006-3097, U.S. Geological Survey, July 2006. http://pubs.usgs.gov/fs/2006/3097/fs2006-3097.pdf.

[110] A.K. Bhuie, O.A. Ogunseitan, J.D.M. Saphores, and A.A. Shapiro. Environmental and economic trade-offs in consumer electronic products recycling: a case study of cell phones and computers. In *Electronics and the Environment, 2004. Conference Record. 2004 IEEE International Symposium on*, pages 74–79, may 2004.

[111] C.S. Tucker and H.M. Kim. Predicting emerging product design trend by mining publicly available customer review data. In *Proceedings of INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN*, pages 43–52, Copenhagen, Denmark, 2011.

[112] R. Rai. Identifying key product attributes and their importance levels from online customer reviews. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2011)*, number DETC2012-70493, 2012.

[113] T. Stone and S.K. Choi. Extracting consumer preference from user-generated content sources using classification. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2013)*, Portland, USA, 2013. DETC2013-13228.

[114] J. Ma and H.M. Kim. Continuous preference trend mining for optimal product design with multiple profit cycles. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2013)*, number DETC2013-12163, Portland, USA, 2013.

[115] J. Ma and H.M. Kim. Continuous preference trend mining for optimal product design with multiple profit cycles. *Journal of Mechanical Design*, 136(6):061002, 2014.

[116] A. Kusiak and M. Smith. Data mining in design of products and production systems. *Annual Reviews in Control*, 31(1):147–156, 2007.

[117] Y. Zhao, V. Pandey, H.M. Kim, and D. Thurston. Varying lifecycle lengths within a product take-back portfolio. *Journal of Mechanical Design*, 132(9):091012, 2010.

[118] J.R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, 1986.

[119] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2005.

[120] K. Cheung, J.T. Kwok, M. . Law, and K. Tsui. Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35(2):231–243, 2003.

[121] N. Archak, A. Ghose, and P.G. Ipeirotis. Deriving the pricing power of product features by mining consumer reviews. *Management Science*, 57(8):1485–1509, 2011.

[122] L. Ferreira, N. Jakob, and I. Gurevych. A comparative study of feature extraction algorithms in customer reviews. In *Semantic Computing, 2008 IEEE International Conference on*, pages 144–151, 2008.

[123] M. Abulaish, Jahiruddin, M.N. Doja, and T. Ahmad. Feature and opinion mining for customer review summarization. In *Proceedings of the 3rd International Conference on Pattern Recognition and Machine Intelligence*, PReMI '09, pages 219–224, Berlin, Heidelberg, 2009. Springer-Verlag.

[124] R. Decker and M. Trusov. Estimating aggregate consumer preferences from online product reviews. *International Journal of Research in Marketing*, 27(4):293–307, 2010.

[125] G. De'ath. Multivariate regression trees: A new technique for modeling species-environment relationships. *Ecology*, 83(4):1105–1117, 2002.

[126] S. Yue, P. Pilon, and G. Cavadias. Power of the mannkendall and spearman's rho tests for detecting monotonic trends in hydrological series. *Journal of Hydrology*, 259(14):254–271, 2002.

[127] R. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008.

[128] J.R. Quinlan. Learning with continuous classes. pages 343–348. World Scientific, 1992.

[129] Y. Wang and I. H. Witten. Inducing model trees for continuous classes. In *Proc. of the 9th European Conf. on Machine Learning Poster Papers*, pages 128–137, 1997.

[130] M. Kwak, H.M. Kim, and D. Thurston. Formulating second-hand market value as a function of product specifications, age, and conditions. *Journal of Mechanical Design*, 134(3), 2012.

[131] D.L. Shrestha and D.P. Solomatine. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks*, 19(2):225–235, 2006.

[132] J. Ma and H.M. Kim. Predictive, data-driven product family design. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2014)*, number DETC2014-34753, Buffalo, USA, 2014.

[133] J. Ma and H.M. Kim. Massive-scale user preference clustering for product family architecture design. In *International Conference on Human Behavior in Design 2014*, ASCONA, SWITZERLAND, 2014.

[134] J. Ma and H.M. Kim. Product family architecture design with predictive, data-driven product family design method, submitted for publication. Research in Engineering Design.

[135] M.M. Tseng. Design for mass customization by developing product family architecture. In *1998 ASME Design for Manufacture Conference*, number DETC98/DFM-5717, Atlanta, GA, September 13-16 1998. American Society of Mechanical Engineers.

[136] W. Chen, C. Hoyle, and H.J. Wassenaar. *Decision-Based Design: Integrating Consumer Preferences into Engineering Design*. SpringerLink : Bücher. Springer, 2012.

[137] D.P. Rutherford and W.E. Wilhelm. Forecasting notebook computer price as a function of constituent features. *Comput. Ind. Eng.*, 37(4):823–845, December 1999.

[138] Philip DesAutels and Pierre Berthon. The PC (polluting computer): Forever a tragedy of the commons? *The Journal of Strategic Information Systems*, 20(1):113–122, 2011.

[139] Wilbert E. Wilhelm, Purushothaman Damodaran, and Jingying Li. Prescribing the content and timing of product upgrades. *IIE Transactions*, pages 647–664, 2003.

[140] P. Damodaran and W. E. Wilhelm. Branch-and-price approach for prescribing profitable feature upgrades. *International Journal of Production Research*, 43(21):4539–4558, 2005.

[141] Minjung Kwak and Harrison Kim. Market positioning of remanufactured products with optimal planning for part upgrades. *Journal of Mechanical Design*, 135(1):011007, 2013.

[142] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability - Vol. 1*, pages 281–297. University of California Press, Berkeley, CA, USA, 1967.

[143] Wei Sun, Junhui Wang, and Yixin Fang. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electron. J. Stat.*, 6:148–167, 2012.

[144] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1–58, March 2009.

[145] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[146] C.B. Do and S. Batzoglou. What is the expectation maximization algorithm? *Nature Biotechnology*, 26(8):897–899, August 2008.

[147] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *In Proceedings of the 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, 2000.

[148] J. Ma and H.M. Kim. Continuous preference trend mining for optimal product design with multiple profit cycles. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2014)*, number DETC2014-34755, Buffalo, USA, 2014.

[149] J. Ma and H.M. Kim. Predictive usage mining for life cycle assessment, submitted for publication. Transportation Research Part D.

[150] Cassandra Telenko and Carolyn C Seepersad. Probabilistic graphical modeling of use stage energy consumption: A lightweight vehicle example. *Journal of Mechanical Design*, 136(10):101403, 2014.

[151] T. Jackson. Analyzing seasonal time series with periodic low volumes. In *Proceedings of International Symposium on Forecasting*, San Diego, USA, 2010.

[152] R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Springer, 2011.

[153] R. Killick, P. Fearnhead, and I.A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[154] Rebecca Killick and Idris A. Eckley. *changepoint: An R package for changepoint analysis*, 2011. R package version 0.5.

[155] M. Goedkoop and S. Spriensma. The eco-indicator 99: A damage oriented method for life cycle impact assessment. Annex report, Pre Consultant, B.V., Jun 2001. http://www.pre-sustainability.com.

[156] Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9):2579–2589, September 2011.

[157] G. Shmueli. To Explain or to Predict? *Statistical Science*, 25(3):289–310, 2010.

[158] J. Ma and H.M. Kim. Predictive modeling of product returns for remanufacturing. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2015)*, number DETC2015-46875, Boston, USA, 2015.